Appendix D - Module Specifications

This appendix contains abbreviated versions of module specifications for the LPS detailed design. Entries appear in alphabetical order. Each abbreviated specification contains the following elements.

- Unit Name – the module's name.
- Purpose - a brief description of the processing the module performs.
- Return Value - a description of the values returned by the module as a function value.
- Parameters - a list of the information passing between the module and its invoker.
- External Variables - a list of the global data items accessed by the module with a description of their use.
- External File/Database Table/Record References - a list of the data files and database tables accessed by the module with a description of their use.
- Assumptions, Constraints, and Restrictions – a description of each, if any apply to the module's design.
- Notes - additional information as required.

Complete module specifications may be reviewed on-line by accessing the detailed design model, LPS_DDS, stored on ipdcase.gsfc.nasa.gov. Please contact TBD for access to the on-line version of the detailed design model.

```
UNIT NAME:      fs_align_frames_n_output

PURPOSE:        to align the found frames and call the output module

PARAMETERS:
RETURN_CODE : control_out
numFrames : data_in
offset : data_in
startThisBuff : data_in
buff : data_in
invertedFlag : data_in
slipped_bits : data_in
flipped_bits : data_in
OutputModulePtr : data_in
state : data_in

RETURN VALUE:
Type = Integer

Value          Description
-------------- -------------------------------------------------------------
SUCCESS         Output module returned successfully
FAILURE         Output module returned unsuccessfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
Frame_Sync      fsControl           TFSPControls    I/O Frame Sync Control Vars

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit fs_align_frames_n_output
*****************************************************************
```

UNIT NAME: fs_frame_sync

PURPOSE: Locates and Byte Synchronizes All of the CADUs in a Raw Data Block.
Note that this is a library module.

PARAMETERS:
RETURN_CODE : control_out
num_bytes : data_in
buff : data_in

RETURN VALUE:
Type = Integer

Value	Description
SUCCESS	Processed and outputted frames successfully
FAILURE	Error outputting frames

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
Frame_Sync	fsControl	TFSPControls	I/O	Frame Sync Control Vars
Frame_Sync	fsDataSize	Unsigned Long	I/O	Size of the Data Buff
Frame_Sync	fsData	BytePtrType	I/O	Data buffer

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
Since fs_frame_sync uses some global routines, only one instance can be running at the same time for one process.

NOTES:
fs_initialize must be called first to set up some data. fs_frame_sync will probably be called many times. Finally, fs_terminate must be called to flush buffers and free memory. This routine when it finds a frame sync pattern automatically outputs the next frame length bytes as a frame. IF the routine goes into flywheel mode it waits until LOCK is reachieved. If lock is never reachieved, no frames are written out.

End Unit fs_frame_sync
****

```
UNIT NAME:              fs_get_stats

PURPOSE:                To retrieve the statistics of the frame synchronization
                        process so far.

PARAMETERS:
stats : data_out

RETURN VALUE:
Type = Void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
Frame_Sync      fsControl           TFSPControls     I  Frame Sync Control Vars

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        This function is not guaranteed if called by the output module.

NOTES:

End Unit fs_get_stats
****************************************************************
```

UNIT NAME: fs_initialize

PURPOSE: Initializes the Frame Synchronization Algorithm

PARAMETERS:
RETURN_CODE : control_out
params : data_in

RETURN VALUE:
Type = Integer

Value	Description
SUCCESS	Successful initialization
FAILURE	Error initializing frame sync

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
Frame_Sync	fsData	BytePtrType	O	Data buffer
Frame_Sync	fsDataSize	Unsigned Long	O	Size of the Data Buff
Frame_Sync	fsControl	TFSPControls	O	Frame Sync Control Vars
Frame_Sync	fsData	Byte[][256]	O	Lookup tables

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
This routine assumes that the input parameters have been checked for validity.

NOTES:

End Unit fs_initialize
****

```
UNIT NAME:      fs_match_fsp

PURPOSE:        This function determines if a frame sync pattern exists at the
                specified byte and bit location.  Bit Flips are handled, but
          bit slips ARE NOT.

PARAMETERS:
buff : data_in
offset : data_in
MaxBitFlips : data_in
flipped_bits : data_out
inverted_flag : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = Boolean

Value          Description
-------------- -------------------------------------------------------------
TRUE            Frame sync pattern was found at location
FALSE           Frame sync pattern was not found at location

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
Frame_Sync      fsControl           TFSPControls     I  Frame Sync Control Vars
Frame_Sync      fsBytes             Byte[][256]      I  Lookup tables

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit fs_match_fsp
****************************************************************
```

```
UNIT NAME:      fs_match_fsp_slip

PURPOSE:        This function determines if a frame sync pattern exists at the
specified
                byte and bit location.  Bit Flips AND bit slips are handled.

PARAMETERS:
buff : data_in
offset : data_inout
mode : data_in
flipped_bits : data_out
slipped_bits : data_out
inverted_flag : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = Boolean

Value          Description
-------------- -------------------------------------------------------------
TRUE            Frame sync pattern was found at location
FALSE           Frame sync pattern was not found at location

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
Frame_Sync      fsControl           TFSPControls     I  Frame Sync Control Vars

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

        This function assumes that buff points to a location in memory that is
        AT LEAST offset - fsControl.input.bitSlipsPerMode back.

End Unit fs_match_fsp_slip
*****************************************************************
```

```
UNIT NAME:      fs_terminate

PURPOSE:        Terminates the Frame Synching Software Gracefully.  Generates
                Some Summary Statistics.

PARAMETERS:
RETURN_CODE : control_out
stats : data_out

RETURN VALUE:
Type = Integer

Value          Description
-------------- ------------------------------------------------------------------
SUCCESS         Terminated successfully
FAILURE         Error in termination

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
Frame_Sync      fsControl           TFSPControls    I/O Frame Sync Control Vars
Frame_Sync      fsDataSize          Unsigned Long   I/O Size of the Data Buff
Frame_Sync      fsData              BytePtrType     I/O Data buffer

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
        Last partial frame is not output.  THE ONLY REASON WHY CALLING ROUTINE
WOULD
        GET A FAILURE BACK IS IF OUTPUT_MODULE RETURNED AN ERROR.

End Unit fs_terminate
*****************************************************************
```

```
UNIT NAME:GetNumErrsPerFrame

PURPOSE: To get the number of header errors corrected by RS EDAC

PARAMETERS:
lpsInt32Type : data_out

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
***************************************************************
```

```
UNIT NAME: GetNumUncorrErrsPerFrame

PURPOSE: To get the number of header errors not corrected by the RS EDAC

PARAMETERS:
lpsInt32Type : data_out

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
****************************************************************
```

UNIT NAME: idp_ACCA

PURPOSE: The main procedure for generating ACCA scores on band files.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
idp_ACCAState	idp_accaSceneScores	{ idpACCAStruct }	O	linked list of scene/ quadrant scores
idp_ACCAState	idp_accaExitFlag	lpsUInt8Type	O	Normal or abnormal termination

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

idp_ACCA will ONLY process subintervals in format 1. It shuts down on LPS_SIG_TERM or the IDP_LAST_PARTIAL message from idp_Band that this is the last scene.

idp_ACCA assumes the data has been radiometrically calibrated if it needs to be.

This function does not return. The idp_ACCAShutdown function does the actual exit.

End Unit idp_ACCA
****

```
UNIT NAME:      idp_ACCAAssessScene

PURPOSE:        To look at the band pixels for the entire scene to determine
                whether the cloud cover score.

PARAMETERS:
RETURN_CODE : control_out
idp_accaFillValueArg : data_in
idp_accaSceneArg : data_out
idp_accaSceneDataArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS     Successful termination
LPS_FAILURE     Termination with failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES: None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_ACCAAssessScene
*****************************************************************
```

UNIT NAME: idp_ACCAInit

PURPOSE: To initialize the ACCA Process

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Successful Initialization
LPS_FAILURE	Initialization failed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
idp_ACCAState	idp_accaSubIntvId	lpsUInt32Type	O	SubInterval being processed
idp_ACCAState	idp_accaMethod	lpsStringType	O	algorihmic method used to calculate cloud cover
idp_ACCAState	idp_accaRatio	lpsUInt8Type	O	Ratio to use when reducing data
idp_ACCAState	idp_accaFillValue	lpsUInt8Type	O	fill value for scene data
idp_ACCAState	idp_accaDBOpenFlag	lpsBooleanType	O	flag to tell if connected to db
idp_ACCAState	idp_accaFileDescrs	{ lpsFilePtrType }	O	File IDs for band files
idp_ACCAState	idp_accaSceneScores	{ idpACCAStruct }	O	Scene Scores Linked List

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_ACCAInit
****

```
UNIT NAME:      idp_ACCASceneCalc

PURPOSE:        To determine the percentage of cloud cover for a scene and its
                four quadrants.  Only processes one scene per invocation.

PARAMETERS:
idp_accaSceneIdArg : data_in
idp_accaSceneArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS     Successful processing of scene
LPS_FAILURE     Unsuccessful processing of scene

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Band_File                  I    Contains the pixel values for all bands

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
idp_ACCAState   idp_accaFillValue   Fill_Value       I  fill value for data,
                                                        from Fill_Value
idp_ACCAState   idp_accaFileDescrs  { Integer }     I/O band file descriptors

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        This function assumes that records are synchronized across all band
files,
        i.e. that record 1 of band 1 has the same exact time stamp of record 1 of
        band 2, etc.

NOTES:

End Unit idp_ACCASceneCalc
*****************************************************************
```

UNIT NAME: idp_ACCAShutdown

PURPOSE: To shutdown and terminate the ACCA Process.

PARAMETERS:

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	idp_ACCA completed subinterval successfully
LPS_FAILURE	idp_ACCA completed subinterval unsuccessfully
LPS_WARNING	idp_ACCA completed subinterval successfully with warnings
IDP_ABORTED_TERMINATION	idp_ACCA was shutdown by an external LPS_SIG_TERM signal

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
idp_ACCAState	idp_accaSubIntvId	lpsUInt32Type	I	Subinterval id
idp_ACCAState	idp_accaFileDescrs	{lpsFilePtrType}	I	Band file descriptors
idp_ACCAState	idp_accaSceneScores	{idpACCAStruct}	I	linked list of scene/ quadrant scores
idp_ACCAState	idp_accaSceneScores	{idpACCAStruct}	O	Scene Scores Linked List
idp_ACCAState	idp_accaExitFlag	lpsInt8Type	I	termination status
idp_ACCAState	idp_accaDBOpenFlag	lpsBooleanType	I	db connected status

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_ACCAShutdown
****

```
UNIT NAME:
        idp_Band

PURPOSE:

        This module generates either three or six band files based on the
        format of the scene data.

PARAMETERS:
idp_bandContactSeqIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value            Description
--------------   ------------------------------------------------------------
LPS_SUCCESS      Exit successfully
LPS_FAILURE      Exit with error

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_Band
*****************************************************************
```

```
UNIT NAME:  idp_BandAppend

PURPOSE:

This module appends the DriftTime and StatusInfo to the band files.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Exit successfully
LPS_FAILURE    Exit with error

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ---------------------------------------------

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
idp_BandState   StatusInfo                           I  Status Info
idp_BandState   DriftTime                            I  Drift Time

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BandAppend
*****************************************************************
```

UNIT NAME: idp_BandClose

PURPOSE:

This module appends data to the band files and closes the message queue

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = VOID

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
idp_BandState	Status		I/O	Band status flag
idp_BandState	FormatId		I	Format Id
idp_BandState	StatusInfo		I	Status Info
idp_BandState	BandFileNames		I	File Names
idp_BandState	BandFileDescr		I	File descriptors
idp_BandState	SubIntvId		I	Subinterval id
idp_BandState	SubIntvSceneNum		I	Scene counter
idp_BandState	SceneCtrScanNum		I	Center Scan number

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

The return value of this signal handler is still an open issue.

End Unit idp_BandClose
****

UNIT NAME: idp_BandCreFiles

PURPOSE:

This module creates the band files and sends a message indicating the name of the band files and whether the data is format 1 or 2.

PARAMETERS:
idp_bandContactSeqIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Exit successfully
LPS_FAILURE	Exit with error

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
idp_BandState	StatusInfo		O	Status Info
idp_BandState	FormatId		I/O	Format Id
idp_BandState	BandFileNames		O	Band file names
idp_BandState	BandFileDescr		O	Band file descriptors
idp_BandState	SubIntvId		I/O	Subinterval id's

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BandCreFiles
****

```
UNIT NAME:  idp_BandDetSubIntv

PURPOSE:  To determine if a scenes worth of data is available for further
          processing

PARAMETERS:
idp_bandFirstTimeThroughArg : data_in
idp_bandSceneCenterArg : data_in
idp_bandMajorFrameTimeArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    Current scene is available for further processing
LPS_NOSCENE    Current scene is not available for further processing

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BandDetSubIntv
*****************************************************************
```

```
UNIT NAME:  idp_BandEndOfScene

PURPOSE:  To determine if the current scene is a full scene or a partial
          scene and to notify ACCA and Browse when a full scene is available.

PARAMETERS:
RETURN_CODE : control_out
idp_bandSceneInfoArg : data_in

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS     No error encountered processing current scene
LPS_FAILURE        Error encountered processing current scene

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
INCLUDE FILE    IDP_MAFTIME          Integer array    I  Array of MAF times
INCLUDE FILE    IDP_NUMMAFTIMES      Integer         I/O Number of MAF times in
                                                            MAF time array
ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BandEndOfScene
*****************************************************************
```

UNIT NAME: idp_BandEndPartial

PURPOSE: To determine if the available major frames do not support the end of the current scene.

PARAMETERS:
RETURN_CODE : control_out
idp_lastTimeIndexArg : data_out
idp_firstTimeIndexArg : data_in
idp_bandSceneStopArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_TRUE	Major frames support the end of the current scene
LPS_FALSE	Major frames do not support the end of the current scene

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
None		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
INCLUDE FILE	IDP_MAFTIMES	TIME	I	Major frame time array
INCLUDE FILE	IDP_NUMMAFTIMES	Integer	I	Number of major frame times in the major frame time array

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BandEndPartial
****

```
UNIT NAME: idp_BandFileDelete

PURPOSE:

This module deletes the band files pertaining to the current subinterval.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = VOID

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BandFileDelete
*************************************************************
```

```
UNIT NAME:  idp_BandFileWrite

PURPOSE  :  To write the major frame bands to the appropriate band files

PARAMETERS:
idp_bandFilesArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE
Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS      Able to write major frame data
LPS_FAILURE    Unable to write major frame data

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
lps_AlignedBands            I  Major frame
BANDFILE1                   O  File containing Band 1 detector data
BANDFILE2                   O  File containing Band 2 detector data
BANDFILE3                   O  File containing Band 3 detector data
BANDFILE4                   O  File containing Band 4 detector data
BANDFILE5                   O  File containing Band 5 detector data
BANDFILE6                   O  File containing Band 6 detector data
BANDFILE7                   O  File containing Band 7 detector data
BANDFILE8                   O  File containing Band 8 detector data

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

NOTES:
01.  The structure BAND_ATTRIBUTE is local to this unit and contains the
     number of detectors and the number of bytes of data associated with
     each band.
02.  The variable HEADEROFFSET contains the number of bytes prior to the
     first byte of data, i.e., the length of the header.

End Unit idp_BandFileWrite
*****************************************************************
```

UNIT NAME: idp_BandFillFile

PURPOSE:

This module places MFPS data from shared memory into individual band files and notifies Browse and ACCA each time a scene has been placed in the file.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Exit successfully
LPS_FAILURE	Exit with error
LPS_END_OF_CONTACT	End of Contact Period

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
idp_BandState	SubIntvSceneNum		O	

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
The module is capable of detecting when a scene or subinterval ends and when a scene is partial or full. Notification to id_Browse and idp_ACCA is made only for format 1 data and to idp_ACCA only for full scenes.

This module is not as detailed as it ought to be. I kept it high level to keep it clear. The detail will be worked out in critical design. There may also be some errors in the logic. This, too, will be worked out in critical design.

End Unit idp_BandFillFile
****

UNIT NAME: idp_BandFindScanNum

PURPOSE:

This module finds the index of the major frame which is closest to the current scene center.

PARAMETERS:
idp_bandSceneCenterArg : data_in
idp_bandMajorFrameIndexArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = VOID

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
idp_BandState	BandScanNum		I/O	Scan number
idp_BandState	Delta			Difference

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

There is a logic problem in that there is no error condition in this module that will handle the situation where a major frame has already been received.

End Unit idp_BandFindScanNum
****

UNIT NAME: idp_BandFindScene

PURPOSE:

This module gets the scene information and drift time from PCDS and calculates the scene start and stop times

PARAMETERS:
RETURN_CODE : control_out
idp_bandMajorFrameTimeArg : data_in
idp_bandSceneCenterArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Scene Center for current scene found
LPS_FAILURE	Exit with error
LPS_NOSCENE	No scene center for current scene found
LPS_END_OF_CONTACT	End of Contact flag

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
idp_BandState				

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

There is a problem with a possible overwrite to perviously found scenes.
This will be reworked in critical design.

End Unit idp_BandFindScene
****

UNIT NAME idp_BandInit:

PURPOSE:

This module initializes the environment for band file generation by setting up a signal handler, creating and attaching to a message queue, attaching to shared memory and connecting to the database.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Initialiation successfully
LPS_FAILURE	Initialization failed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
idp_BandState	PCDDriftFifoId	Int	O	Message queue id
idp_BandState	PCDSceneFifoId	Int	O	Message queue id
idp_BandState	ACCAFifoId	Int	O	Message queue id
idp_BandState	BrowseFifoId	Int	O	Message queue id
idp_BandState	MFPShMemId	Int	O	Message queue id
idp_BandState	BandScanNum	Int	O	Scan number
idp_BandState	Status			Band state

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BandInit
****

```
UNIT NAME:  idp_BandReadMF

PURPOSE:  To read a major frame from shared memory and to write each band
             to its band file.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS      Able to read and process the major frame
LPS_FAILURE    Unable to read and process the major frame

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
INCLUDE FILE    lps_AlignedBands     Structure        I  Major frame

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BandReadMF
*****************************************************************
```

```
UNIT NAME: idp_BandSceneTimes

PURPOSE:  To calculate the scene start and stop times.

PARAMETERS:
RETURN_CODE : control_out
idp_bandSceneStopArg : data_out
idp_bandSceneStartArg : data_out
idp_bandSceneInfoArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS      Able to calculate scene start and stop time
LPS_FAILURE    Unable to calculate scene start and stop time

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BandSceneTimes
*****************************************************************
```

```
UNIT NAME:  idp_BandStartPartial

PURPOSE:  To determine if the available major frames do not support the
          beginning of the current scene.

PARAMETERS:
RETURN_CODE : control_out
idp_firstTimeIndexArg : data_out
idp_bandSceneStartArg : data_in

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_TRUE       Major frames support beginning of scene
LPS_FALSE      Major frames do not support beginning of scene

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
INCLUDE FILE    IDP_MAFTIME          TIME             I  Major frame time array
INCLUDE FILE    IDP_NUMMAFTIMES      Integer          I  Number of major frame
                                                           times in the major
                                                           frame time array

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BandStartPartial
*****************************************************************
```

UNIT NAME: idp_BandStatusInfo

PURPOSE:

This module extracts the status information from shared memory and further extracts the format identifier from the status information. It also extracts the subinterval identifier. These values are then placed in the global idp_BandState

PARAMETERS:

RETURN VALUE:
Type = VOID

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
idp_BandState	FormatId		O	Format id
idp_BandState	idp_subIntvId		O	Subinterval id
idp_BandState	StatusInfo		O	Status Info
lps_AllignedBands			I	Alogned Bands

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BandStatusInfo
****

```
UNIT NAME:      idp_Browse

PURPOSE:        The main procedure for generating Browse files from the
                band files

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    successful browse generation processing
LPS_FAILURE    unsuccessful browse generation processing

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
idp_BrowseState idp_browseExitFlag  lpsInt8Type      O  Status of the process
idp_BrowseState idp_browseFileList  { lpsStringType }O  Linked list of multiband
                                                        browse filenames

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

        idp_Browse will ONLY process subintervals in format 1.  It shuts down on
        LPS_SIG_TERM or the IDP_LAST_PARTIAL message from idp_Band that this is
        the last scene.

        idp_Browse assumes the data has been radiometrically calibrated if it
needs
        to be.

        This function does not return.  The idp_BrowseShutdown function does the
        actual exit.

End Unit idp_Browse
*****************************************************************
```

```
UNIT NAME:      idp_BrowseCreateFile

PURPOSE:        To create the multiband browse file and write in the header
                information.

PARAMETERS:
idp_browseSceneIdArg : data_in
idp_browseFilenameArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Create Browse File successfully
LPS_FAILURE    Error creating browse file

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ---------------------------------------------
Browse_File                O  Multiband browse file

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
idp_BrowseState idp_browseMultiDescr lpsFilePtrType  O  file descriptor for
                                                        multiband browse file
idp_BrowseState idp_browseSubIntvId  lpsUIntType     I  Subinterval being
                                                        processed

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BrowseCreateFile
*****************************************************************
```

```
UNIT NAME:  idp_BrowseInit

PURPOSE:        To initialize the Browse process

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE
Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    successful Initialization
LPS_FAILURE    Initialization with errors

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
idp_BrowseState idp_browseSubIntvId lpsUIntType      O  SubInterval being
                                                        processed
idp_BrowseState idp_browseMulti1    lpsUInt8Type     O  Band file to use for
                                                        first multiband
idp_BrowseState idp_browseMulti2    lpsUInt8Type     O  Band file to use for
                                                        second multiband
idp_BrowseState idp_browseMulti3    lpsUInt8Type     O  Band file to use for
                                                        third multiband
idp_BrowseState idp_browseSubs      lpsUInt8Type     O  Subsampling Ratio to use
when
                                                        reducing data
idp_BrowseState idp_browseWave      lpsUInt8Type     O  Wavelet Ratio to use when
                                                        reducing data
idp_BrowseState idp_browseFillValue lpsUInt8Type     O  fill value for scene data
idp_BrowseState idp_browseDBOpenFlag lpsBooleanType  O  flag to tell if connected
                                                        to db
idp_BrowseState idp_browseFileList  { lpsStringType }0  Linked list of multiband
                                                        browse filenames
idp_BrowseState idp_browseFileDescrs { lpsFilePtrType }O File IDs for band files

End Unit idp_BrowseInit
****************************************************************
```

```
UNIT NAME:      idp_BrowseSceneProc

PURPOSE:        To process one scene to create Browse file

PARAMETERS:
idp_browseFilenameArg : data_out
idp_browseSceneIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS     Successful termination
LPS_FAILURE     Termination with failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
idp_BrowseState idp_browseFileDescrs { lpsFilePtrType } I Band File Descriptors

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BrowseSceneProc
*****************************************************************
```

```
UNIT NAME:  idp_BrowseShutdown

PURPOSE:        To shutdown and terminate the idp_Browse process

PARAMETERS:

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS     idp_Browse completed subinterval successfully
LPS_FAILURE     idp_Browse completed subinterval unsuccessfully
LPS_WARNING     idp_Browse completed subinterval successfully with warnings
IDP_ABORTED_TERMINATION idp_Browse was shutdown by an external LPS_SIG_TERM
                signal

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Browse_File                N/A  Multiband browse file

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
idp_BrowseState idp_browseReduced   lpsUInt8Type     I  Memory for reduced image
idp_BrowseState idp_browseExitFlag  lpsUIntType      I  Status of idp_Browse
idp_BrowseState idp_browseFileList  { lpsStringType }I  Linked list of multiband
                                                        browse files
idp_BrowseState idp_browseSubIntvId lpsUIntType      I  Subinterval just
processed
idp_BrowseState idp_browseMultiDescr lpsFilePtrType  I  Current Multiband file
descr
idp_BrowseState idp_browseFileDescrs { lpsFilePtrType }I Band File Descriptors
idp_BrowseState idp_browseDBOpenFlag lpsBooleanType  I  Whether DB is open

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_BrowseShutdown
*****************************************************************
```

```
UNIT NAME:      idp_BrowseSubs

PURPOSE:        Reduce the raw image data for a scene by subsampling

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    Subsampled successfully
LPS_FAILURE    Error subsampling

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Band_File                   I  Band file pixels to subsample

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
idp_BrowseState idp_browseReduced { lpsUInt8Type }   O  Reduced Browse image
idp_BrowseState idp_browseMulti1    lpsUInt8Type     I  Band # for multiband
idp_BrowseState idp_browseMulti2    lpsUInt8Type     I  Band # for multiband
idp_BrowseState idp_browseMulti3    lpsUInt8Type     I  Band # for multiband
idp_BrowseState idp_browseSubs      lpsUInt8Type     I  Subsampling Ratio to use
when
                                                        reducing data
idp_BrowseState idp_browseFileDescrs { lpsFilePtrType }I File IDs for band files

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        This routine reduces N number of scanlines from the raw image
        where N is always divisible by idp_browseSubs plus extra
        scanlines

NOTES:

End Unit idp_BrowseSubs
*****************************************************************
```

```
UNIT NAME:  idp_BrowseWave

PURPOSE:        To reduce subsampled image data by wavelets

PARAMETERS:
idp_browseFilenameArg : data_out
idp_browseSceneIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE
Value          Description
-------------- -------------------------------------------------------------------
LPS_SUCCESS    Exit successfully
LPS_FAILURE    Exit with error

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Browse_File                O    Multiband Browse file for the scene

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
idp_BrowseState idp_browseReduced   { lpsUInt8Type } I/O Memory for reduced
browse
idp_BrowseState idp_browseWave      lpsUInt8Type     I  Wavelet Ratio
idp_BrowseState idp_browseMultiDescr lpsFilePtrType I/O File Descr for multiband
                                                         browse file

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        It's assumed that idp_browseWave is a power of 2 (2,4,8,or 16) which
        controls the number of iterations of wavelets performed (1,2,3,4).

NOTES:  idp_browseReduced contains the subsampled image when this routine is
              entered. Then the wavelet reduction algorithm is applied some
amount of           times. After each iteration, idp_browseReduced contains
idp_browseReduced
        reduced by a fourth.

End Unit idp_BrowseWave
*****************************************************************
```

```
UNIT NAME:      idp_ChildFindRecord

PURPOSE:        Find the record in the band files that contains the time
                specified

PARAMETERS:
RETURN_CODE : control_out
idp_cfrTimeArg : data_in
idp_cfrFileDescrsArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS     Successful termination
LPS_FAILURE     Termination with failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Band_File                  I    Contains the pixel values for all bands

EXTERNAL VARIABLES: None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
        This routine is TBD for the seek because of the HDF peculiarities.
        After this routine finishes, the files will be pointing to the
        correct record.

End Unit idp_ChildFindRecord
*****************************************************************
```

```
UNIT NAME: idp_ChildInit

PURPOSE:   To initialize an IDP Child Process

PARAMETERS:
idp_childListArg : data_out
idp_childShutdownArg : data_in
idp_childFifoKeyArg : data_in
idp_childFileDescrsArg : data_out
idp_childDBOpenFlagArg : control_out
idp_childFillValueArg : data_out
idp_childSubIntvIdArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS     Successful Initialization
LPS_FAILURE     Initialization failed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Band_File                  N/A   Band Files to open for read

EXTERNAL VARIABLES: None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_ChildInit
****************************************************************
```

```
UNIT NAME:      idp_ChildShutdown

PURPOSE: To shutdown and terminate the Child Process.

PARAMETERS:
idp_childListArg : data_in
idp_childFileDescrsArg : data_in
idp_childDBOpenFlagArg : control_in
idp_childFifoKeyArg : data_in
idp_childExitFlagArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS     Successful termination
LPS_FAILURE     Termination with failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Band_File                 N/A   Band Files to close

EXTERNAL VARIABLES: None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_ChildShutdown
*****************************************************************
```

```
UNIT NAME:  idp_db_bandUpdate

PURPOSE:
        To update band file names, bands present, scan number of scene center,
        and scene count for each subinterval into IDP_Acct database table

PARAMETERS:
idp_bandsPresentArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Successfully updated to the database table
LPS_FAILURE    Failed to update to the database table

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
idp_BandState   BandFileNames                        I
idp_BandState   SubIntvSceneNum                      I
idp_BandState   SceneCtrScanNum                      I

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The caller has connected to the database server prior to this call
NOTES:

There MUST be a lock on the database during updates and inserts, otherwise
there is a possible multiple insert problem.

End Unit idp_db_bandUpdate
*****************************************************************
```

```
UNIT NAME:      idp_db_GetFillValue

PURPOSE:        To retrieve the band data fill value from the Valid_MFP_Parms
                table.

PARAMETERS:
RETURN_CODE : control_out
idp_db_gfvFillValueArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS     Successful completion
LPS_FAILURE     Failed to complete successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -------------------------------------------------
Valid_MFP_Parms            I  Valid MFP Parameters (contains fill value)

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_db_GetFillValue
*****************************************************************
```

```
UNIT NAME:  idp_db_GetIDPParms

PURPOSE:

This modules retrieves the following values from the database table
Valid_Band_Parms:  Wave, Subs, Mono, Multi1, Multi2, Multi3, cca_method,
and cca_ratio.

PARAMETERS:
idp_db_gipParmsArg : data_inout
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Exit successfully
LPS_FAILURE    Exit with error

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Valid_Band_Parms            I    Valid Band Parameters

EXTERNAL VARIABLES: None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_db_GetIDPParms
*****************************************************************
```

```
UNIT NAME:
        idp_db_InsertScores

PURPOSE:
        To insert the acca scores for all scenes and the quadrants in a
        subinterval into the IDP_Acct table

PARAMETERS:
idp_db_isSubIntvIdArg : data_in
idp_db_isSceneScoreArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS     Successful insertion
LPS_FAILURE     insertion with failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
IDP_Acct                    O    IDP Accounting Table

EXTERNAL VARIABLES: None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        This routine assumes that the scene records have already been inserted
        and committed into the IDP_Acct table.

NOTES:

End Unit idp_db_InsertScores
*****************************************************************
```

```
UNIT NAME:
        idp_Main

PURPOSE:
        IDPS main routine

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Contact period processing successful
LPS_FAILURE    Unsuccessful processing

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
idp_MainState   idp_mainChild       idp_mainChildStruct
                                                    I/O Different Children used
                                                        by idp_Main
idp_MainState   idp_mainExitFlag    lpsInt8Type      O  Normal or abnormal
termination
idp_MainState   idp_mainLastSubIntvFlag
                                    lpsBooleanType  I/O Last Subinterval?

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

        This function does not return.  The idp_MainShutdown routine actually
        performs the exit.

End Unit idp_Main
*****************************************************************
```

```
UNIT NAME:
        idp_MainInit

PURPOSE:
        To initialize IDPS main process

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------------
LPS_SUCCESS    Initialiation successful
LPS_FAILURE    Initialization failed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
idp_MainState   idp_mainChild       idpMainChildStruct
                                                     O Different Children used
                                                       by idp_Main

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_MainInit
*****************************************************************
```

```
UNIT NAME:
        idp_MainProcessChildSignal

PURPOSE:
        To wait on child exit and report exit status event

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = Void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
idp_MainState   idp_mainChild       idpMainChildStruct I/O Different Children
used
                                                       by idp_Main
idp_MainState   idp_mainExitFlag    lpsInt8Type         O  State of the process
idp_MainState   idp_mainLastSubIntvFlag lpsBooleanType  O  Last Subinterval or
not?

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_MainProcessChildSignal
****************************************************************
```

```
UNIT NAME:      idp_MainShutdown

PURPOSE:        To shutdown and terminate the main IDP process

PARAMETERS:

RETURN VALUE:
Type = Void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
idp_MainState   idp_mainChild       idpMainChildStruct I Different Children used
                                                          by idp_Main
idp_MainState   idp_mainExitFlag    lpsInt8Type        I Normal or abnormal
                                                         termination

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit idp_MainShutdown
*****************************************************************
```

```
UNIT NAME:  ldt_accept_client

PURPOSE:
  Accept client socket connection.

PARAMETERS:
ldt_acliSockArg : data_in
ldt_acliSockFdArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
****************************************************************
```

```
UNIT NAME:  ldt_authent_req

PURPOSE:
  Authenticate a user requesting a connection.

PARAMETERS:
ldt_authSockFdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
****************************************************************
```

```
UNIT NAME:  ldt_broadcast

PURPOSE:
  Write to a socket.

PARAMETERS:
ldt_broaSdArg : data_in
ldt_broaSockBuffArg : data_in
ldt_broaSizeArg : data_in
ldt_broaObjectTypeArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ---------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
****************************************************************
```

```
UNIT NAME:
  ldt_CheckTimeouts

PURPOSE:
  Logs to LPS journal the contact ID & version ID of any file set
  whose associated DAN has timed out waiting for a DDN.

PARAMETERS:
lps_OSReturnCode : control_out

RETURN VALUE:
Type = LPS_OS_RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_OS_FAILURE Error connecting to or accessing database
LPS_OS_SUCCESS No error connecting to or accessing database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -------------------------------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
  1.  This program will be called by the operating system as part of a
      crontab.
  2.  It is assumed that to conform with unix scripts, LPS_OS_FAILURE
      will be equal to -1 and LPS_OS_SUCCESS will be equal to 0.

NOTES:

  1.  This program could be run 1, 2, or 3 times a day.

End Unit ldt_CheckTimeouts
*****************************************************************
```

```
UNIT NAME: ldt_create_client

PURPOSE: open client socket to specified host/host.

PARAMETERS:
ldt_ecs_host_id : data_in
ldt_port_num : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ----------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
  1. This is a reuse unit, its PDL is not supplied at this time.

End Unit "Name"
****************************************************************
```

```
UNIT NAME:  ldt_create_server

PURPOSE:
  Create a socket and listen for connections.

PARAMETERS:
ldt_csrvPortNumberArg : data_out
ldt_csrvSockArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type =

Value          Description
-------------- ------------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
****************************************************************
```

```
UNIT NAME: ldt_CreateDAN

PURPOSE: To create a DAN.

PARAMETERS:
ldt_sdDANArg : data_out
ldt_sdFileVersionIdArg : data_in
ldt_sdContactSeqIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_FAILURE    ldt_CreateDAN failed to complete
LPS_SUCCESS    ldt_CreateDAN completed successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ---------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit  ldt_CreateDAN
*****************************************************************
```

```
UNIT NAME:
  ldt_db_AccDANXferState

PURPOSE:
  Return indication of whether or not DAN transfer (to the LP DAAC) is
  enabled.

PARAMETERS:
ldt_xferStateArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_FAILURE    Fatal error(s) detected during processing.
LPS_SUCCESS    No fatal error(s) during processing.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
LDT_DAN_Transfer_State      I  DAN transfer state table

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_AccDANXferState
*****************************************************************
```

```
UNIT NAME: ldt_db_CreateDANInfo

PURPOSE: Create a new database table entry for DAN.

PARAMETERS:
ldt_sdSubIntvArg : data_in
ldt_DAN_seqno : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_FAILURE    ldt_db_CreateDANInfo failed to complete
LPS_SUCCESS    ldt_db_CreateDANInfo completes successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -----------------------------------------------
LDT_DAN_INFO               O  Database table containing DAN sequence number and
                              the subinterval IDs the DAN is related to.

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_CreateDANInfo
*****************************************************************
```

```
UNIT NAME: ldt_db_ExtrDANStruct

PURPOSE: To construct the DAN product information structure which will be
         used for building a DAN.

PARAMETERS:
ldt_sdSubIntvArg : data_in
ldt_sdFileVersionIdArg : data_in
ldt_sdContactSeqIdArg : data_in
ldt_dpiptr : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_FAILURE    ldt_db_ExtrDANInfo failed to complete
LPS_SUCCESS    ldt_db_ExtrDANInfo completed successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LPS_File_Info               I  Database table containing files for a file group
Sub_Intv                    I  Database table containing subinterval info
LPS_Configuration           I  Database table containing LPS configuration

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_ExtrDANStruct
*****************************************************************
```

```
UNIT NAME:
  ldt_db_GenftsReport

PURPOSE:
  Format and output LPS file transfer summary report.

PARAMETERS:
ldt_rptDiskUsedArg : data_in
ldt_rptDiskFreeArg : data_in
ldt_rptStopTimeArg : data_in
ldt_rptStartTimeArg : data_in

RETURN VALUE:
Type = VOID

Value          Description
-------------- ------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Sub_Intv                   I  db table containing Subinterval information
LDT_File_Set_Info          I  db table containing LDT file info per Cont seq
                                ID/file ver ID pair
LDT_File_Group_Info        I  db table containing LDT file subinterval info
LPS_File_Info              I  db table containing LDT file info in a subinterval

EXTERNAL VARIABLES:

NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. This unit uses the Oracle report writer to output the LPS file transfer
   summary report.
2. Oracle report writer automatically connect and disconnect the database.

NOTES:

End Unit ldt_db_GenftsReport
*************************************************************
```

```
UNIT NAME:
  ldt_db_GetContactFileNames

PURPOSE:
  Return a list of names of files which are contained in the file set
  identified by the given contact sequence ID / file version ID pair

PARAMETERS:
RETURN_CODE : control_out
ldt_gcfnContactFileNamesArg : data_out
ldt_gcfnFileVersionIdArg : data_in
ldt_gcfnContactSeqIdArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_FAILURE    No data found or error accessing database
LPS_SUCCESS    Data found and no errors accessing database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Sub_Intv table              I  Maps contacts/versions to subintervals
LPS_File_Info table         I  Information about LPS L0R output files

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_GetContactFileNames
*****************************************************************
```

```
UNIT NAME:
  ldt_db_GetIngestedFiles

PURPOSE:
  For the file set identified by the given contact sequence ID and file
  version ID, get a list of names of files which have been ingested,
  i.e., whose file groups have non-NULL DDN times.

PARAMETERS:
RETURN_CODE : control_out
ldt_gifIngestedFilesArg : data_out
ldt_gifFileVersionIdArg : data_in
ldt_gifContactSeqIdArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS        Data found and no errors accessing database
LPS_FAILURE        No data found or error accessing database
LDT_FILE_NOT_FOUND No ingested files found for identified file set

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ---------------------------------------------
Sub_Intv table             I   LPS subinterval table
LDT_File_Group_Info table  I   Contains information on LDTS file groups
LPS_File_Info table        I   Contains information on LPS output files

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_GetIngestedFiles
****************************************************************
```

```
UNIT NAME:
  ldt_db_GetRetentionState

PURPOSE:
  Return an indication of whether or not the file set identified by the
  given contact sequence ID and file version ID is marked for retention.

PARAMETERS:
RETURN_CODE : control_out
ldt_getretRetentionStateArg : data_out
ldt_getretContactSeqIdArg : data_in
ldt_getretFileVersionIdArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ---------------------------------------------------------------
LPS_FAILURE    No data found or error accessing database
LPS_SUCCESS    Data found and no errors accessing database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------------
LDT_File_Set_Info table    I   Contains state information on LDTS file sets

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_GetRetentionState
****************************************************************
```

```
UNIT NAME: ldt_db_GetServerInfo

PURPOSE: Retrieve host id, user id, port number and password from database
         for authentication request.

PARAMETERS:
ldt_port_num : data_out
ldt_passwd : data_out
ldt_user_id : data_out
ldt_ecs_host_id : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_FAILURE    ldt_db_GetServerInfo failed to complete
LPS_SUCCESS    ldt_db_GetServerInfo completes successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LPS_Configuration           I  LPS configuraiton info

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_GetServerInfo
****************************************************************
```

```
UNIT NAME: ldt_db_GetSubintv

PURPOSE: To get subintervals for a contact sequence id/file version id pair.

PARAMETERS:
RETURN_CODE : control_out
ldt_sdSubIntvArg : data_out
ldt_sdContactSeqIdArg : data_in
ldt_sdFileVersionIdArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    ldt_db_GetSubintv completed
LPS_FAILURE    ldt_db_GetSubintv failed to complete

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
Sub_Intv                    I  Subinterval information

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_GetSubintv
*****************************************************************
```

```
UNIT NAME:
  ldt_db_GetSuspDANs

PURPOSE:
  Retrieve a list of contact sequence ID / file version ID pairs each
  identifying a file set in the database which has a suspended DAN.

PARAMETERS:
ldt_sdFileSetListArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ----------------------------------------------------------------
LPS_FAILURE    Get suspended DANs failed to complete.
LPS_SUCCESS    Get suspended DANs completed.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LDT_File_Set_Info          I   db table which contains DAN and file info for each
                               contact sequence ID/file version ID pair

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
- Assumes calling unit has allocated all storage space for output arguments

NOTES:

End Unit ldt_db_GetSuspDANs
*****************************************************************
```

```
UNIT NAME:
  ldt_db_GetTimedOutDANs

PURPOSE:
  Return a list of structures representing timed out DAN's.

PARAMETERS:
RETURN_CODE : control_out
ldt_gettodTimeOutListArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_FAILURE    Error getting current time or error accessing database
LPS_SUCCESS    No error getting current time and no errors accessing
                 database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
LDT_File_Set_Info table   I   Contains state information on LDTS file sets

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

1.  This unit returns a list of structures, each of which contains a
    contact ID, version ID, and timeout value.  The contact ID and
    version ID identify a file set whose pending DAN has timed out
    waiting for a DDN.  Each timeout value is the date/time at which
    the DAN timed out.  The list is populated with contact ID/version
    ID/next timeout triples for all contacts which have not yet been
    ingested and whose next timeout values are non-null.

End Unit ldt_db_GetTimedOutDANs
*****************************************************************
```

```
UNIT NAME:
  ldt_db_InsStateInfo

PURPOSE:
  Insert LDT information into database for a file set from a L0R processing
  for a contact sequence ID and file version ID pair.

PARAMETERS:
ldt_sdFileVersionIdArg : data_in
ldt_sdContactSeqIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_FAILURE    ldt_db_InsStateInfo failed to complete.
LPS_SUCCESS    ldt_db_InsStateInfo completed successfully.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LDT_File_Set_Info           O  LDT information for a file set.

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. It's assumed that ORACLE will validate arguments.

NOTES:

End Unit ldt_db_InsStateInfo
*****************************************************************
```

```
UNIT NAME:
  ldt_db_SetLastXmitTime

PURPOSE:
  Record time a DAN was transmitted in the database.

PARAMETERS:
ldt_sdFileVersionIdArg : data_in
ldt_sdContactSeqIdArg : data_in
ldt_nasaTimeArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_FAILURE    ldt_db_SetLastXmitTime failed to complete.
LPS_SUCCESS    ldt_db_SetLastXmitTime completed successfully.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LDT_File_Set_Info           O  LDT information for a file set.

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_SetLastXmitTime
****************************************************************
```

```
UNIT NAME:
  ldt_db_SetTimeout

PURPOSE:
  Update database with a time at which the pending DAN for the file
  set identified by the given contact sequence ID / file version ID
  pair will "time out", i.e., require notification be sent to the LPS
  journal indicating that a DDN was not received for the DAN in the
  expected period of time.

PARAMETERS:
ldt_settoNextTimeoutArg : data_in
ldt_settoFileVersionIdArg : data_in
ldt_settoContactSeqIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_FAILURE    Error writing to database
LPS_SUCCESS    No errors writing to database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LDT_File_Set_Info table    O   Contains state information on LDTS file sets

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_SetTimeout
*****************************************************************
```

```
UNIT NAME:
  ldt_db_UpdRetentionState

PURPOSE:
  Mark for retention the file set identified by the given contact
  sequence ID and file version ID.

PARAMETERS:
RETURN_CODE : control_out
ldt_updretFileVersionIdArg : data_in
ldt_updretContactSeqIdArg : data_in
ldt_updretRetainArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ----------------------------------------------------------------
LPS_FAILURE    Error writing to database
LPS_SUCCESS    No error writing to database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------------
LDT_File_Set_Info table    O   Contains state information on LDTS file sets

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_UpdRetentionState
*****************************************************************
```

```
UNIT NAME:
  ldt_db_UpdSuspState

PURPOSE:
  Update DAN_suspended status in db for a contact / file version pair.

PARAMETERS:
ldt_suspendArg : data_in
ldt_sdFileVersionIdArg : data_in
ldt_sdContactSeqIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_FAILURE    ldt_db_UpdSuspState failed to complete.
LPS_SUCCESS    ldt_db_UpdSuspState completed.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LDT_File_Set_Info          O   db table which contains DAN and file info for each
                               contact sequence ID/file version ID pair
EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_db_UpdSuspState
*****************************************************************
```

```
UNIT NAME:
  ldt_db_verifyLongDDN

PURPOSE:
  Parse contents of DDN, update receipt times for successful file
  groups; log errors for unsuccessful files.

PARAMETERS:
RETURN_CODE : control_out
ldt_vlddBufArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Parsing was successful
LPS_FAILURE    Message in input buffer is not a long DDN or has an
                 inappropriate length for a long DDN.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LDT_DAN_Info table         I/O Maps DAN sequence numbers to subinterval
                                 ID's
Sub_Intv table             I   Maps contact sequence ID's and file
                                 version ID's with subinterval ID's
LDT_File_Set_Info table    O   Contains state information about LDT file
                                 sets
LDT_File_Group_Info table  O   Contains state information about LDT file
                                 groups
LPS_File_Info table        I   Contains information about individual LPS
                                 output files

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

1.  This unit has been reused from DDF Consumer Simulator code.  Some
    of the PDL has been changed to conform to LPS standards.  Some
    has not.

End Unit ldt_db_verifyLongDDN
*****************************************************************
```

```
UNIT NAME:
  ldt_db_verifyShortDDN

PURPOSE:
  Parse a short DDN.  Update receipt times for all file groups if
  successful DDN; otherwise report errors.

PARAMETERS:
RETURN_CODE : control_out
ldt_vsddBufArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Message in input buffer was parsable.
LPS_FAILURE    Message in input buffer was not parsable.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
LDT_DAN_Info table        I/O Maps DAN sequence numbers to subinterval
                              ID's
Sub_Intv table            I   Maps contact sequence ID's and file
                              version ID's with subinterval ID's
LDT_File_Set_Info table   O   Contains state information about LDT file
                              sets
LDT_File_Group_Info table O   Contains state information about LDT file
                              groups
LPS_File_Info table       I   Contains information about individual LPS
                              output files

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

1.  This unit has been reused from DDF Consumer Simulator code.  Some
    of the PDL has been changed to conform to LPS standards.  Some
    has not.

End Unit ldt_db_verifyShortDDN
****************************************************************
```

```
UNIT NAME:  ldt_DeleteFiles

PURPOSE:
  Delete files in the file set identified by the input contact
  sequence ID and file version ID as indicated by the boolean input
  arguments.

PARAMETERS:
ldt_dlfOnlyDeleteIngestedFilesArg : control_in
RETURN_CODE : control_out
ldt_dlfFileVersionIdArg : data_in
ldt_dlfContactSeqIdArg : data_in
ldt_dlfOverrideRetentionArg : control_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_FAILURE    Unable to access database
LPS_SUCCESS    No error accessing database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

  1.  If the override retention argument is false, this function will
      not delete files marked for retention.
  2.  If the deleted only ingested files argument is true, this
      function will not delete files which have not been successfully
      transferred.
  3.  The LDTS DDN server will always call this function with override
      retention = false and delete only ingested files = true.

End Unit ldt_DeleteFiles
*****************************************************************
```

```
UNIT NAME:
  ldt_delRemoveFiles

PURPOSE:
  Remove output files in the file set identified by the given contact
  sequence ID and file version ID from non-volatile storage.

PARAMETERS:
ldt_rmfOnlyDeleteIngestedFilesArg : control_in
RETURN_CODE : control_out
ldt_rmfContactSeqIdArg : data_in
ldt_rmfFileVersionIdArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_FAILURE    Error accessing database
LPS_SUCCESS    No errors accessing database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LDT_File_Set_Info table       State info on all output files based on
                                contact sequence ID and file version ID
LPS_File_Info table           State info for each individual output
                                file

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

1.  This function will not return unsuccessful status if there are
    problems deleting individual files.  Problems deleting individual
    files will simply be logged to the LPS journal.  However, this
    function will return unsuccessful status if there are problems
    with database access.

NOTES:

End Unit ldt_delRemoveFiles
*****************************************************************
```

```
UNIT NAME: ldt_diff_time

PURPOSE: Get time difference between two input times.

PARAMETERS:
ldt_nasa_diff : data_out
ldt_nasa1 : data_in
ldt_nasa2 : data_in

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ----------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
  1. This is a reuse unit, PDL is not supplied at this time.

End Unit "Name"
****************************************************************
```

```
UNIT NAME:
  ldt_do_action

PURPOSE:
  To call the appropriate ddn verification routine DDA broadcast
  procedure.

PARAMETERS:
ldt_dactSockBuffArg : data_in
ldt_dactMsgTypeArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_FAILURE    Input buffer does not contain a DDN.
LPS_SUCCESS    Input buffer contains a DDN.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

1.  This unit has been reused from DDF Consumer Simulator code.  Some
    of the PDL has been changed to conform to LPS standards.  Some
    has not.
2.  A short DDA is sent in response to either a long DDN or a short DDN.

End Unit ldt_do_action
****************************************************************
```

```
UNIT NAME: ldt_establish_client

PURPOSE: To establish client connection.

PARAMETERS:
RETURN_CODE : control_out
ldt_socket_ptr : data_out
ldt_user_id : data_in
ldt_passwd : data_in
ldr_socket_buff : data_in
ldt_ecs_host_id : data_in

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
  1. This is a reuse unit, its PDL is not supplied at this time.

End Unit "Name"
****************************************************************
```

```
UNIT NAME:
  ldt_GenFTS

PURPOSE:
  Generate LPS file transfer summary.

PARAMETERS:
ldt_gftsStopTimeArg : data_in
ldt_gftsStartTimeArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ---------------------------------------------------------------
LPS_FAILURE    Generate file transfer summary failed to complete.
LPS_SUCCESS    Generate file transfer summary completed.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. Set report start time to midnight of current day, report stop time to current
   time if either (or both) input time is (are) missing.
2. This unit provides the data needed by the Oracle report writer to
   generate the LPS file transfer summary.

NOTES:

End Unit ldt_GenFTS
*****************************************************************
```

```
UNIT NAME:
  ldt_GftsDiskUsage

PURPOSE:
  Generate LPS on-line disk usage information.

PARAMETERS:
ldt_gftsDiskUsedArg : data_out
ldt_gftsDiskFreeArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_FAILURE    Get disk usage failed to complete.
LPS_SUCCESS    Get disk usage completed.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. This unit provides the on-line storage information needed by the the LPS file
   transfer summary.

NOTES:

End Unit ldt_db_GftsDiskUsage
*****************************************************************
```

```
UNIT NAME:
  ldt_input_defaults

PURPOSE:
  Read from a file the DDN server parameters.

PARAMETERS:
RETURN_CODE : control_out
ldt_idefDDNParmsArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_FAILURE    Error allocating memory for temporary buffer
               or error opening parameters file
               or error parsing PVL statements
               or error assigning PVL statement values to output
                  structure
LPS_SUCCESS    Parameters successfully loaded from the file

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ---------------------------------------------
ldt_DDNParameters file     I   Contains parameters needed by the DDN
                               server.

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
1.  This unit has been reused from DDF Consumer Simulator code.  Some
    of the PDL has been changed to conform to LPS standards.  Some
    has not.

End Unit ldt_input_defaults
*****************************************************************
```

```
UNIT NAME: ldt_longdaa

PURPOSE: To process the long daa.

PARAMETERS:
ldt_msg_type : data_in
ldt_bufaddr : data_in
ldt_msg_length : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =

Value          Description
-------------- ------------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
  1. This is a reuse unit, PDL is not supplied at this time.

End Unit "Name"
****************************************************************
```

```
UNIT NAME:  ldt_nasa_time_now

PURPOSE:
  Get the current nasa standard time.

PARAMETERS:
RETURN_CODE : control_out
ldt_nasaTimeArg : data_out

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ----------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
  1. This is a reuse unit, PDL is not supplied at this time.

End Unit "Name"
*************************************************************
```

```
UNIT NAME:  ldt_parse_buffer

PURPOSE:
  This unit is part of the PVL tools code.

PARAMETERS:
ldt_prseBufferArg : data_in
ldt_prseLengthArg : data_in
RETURN_CODE : control_out
ldt_prseStatusArg : control_out

RETURN VALUE:
Type =

Value          Description
-------------- ------------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
****************************************************************
```

```
UNIT NAME:  ldt_prepare_short_DDA

PURPOSE:
  Prepare a short DDA to send in response to a short or long DDN.

PARAMETERS:
ldt_psddDispositionArg : data_in
ldt_psddDDAArg : data_out

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
****************************************************************
```

```
UNIT NAME: ldt_RcvDaa

PURPOSE: To receive and process a Data Availability Acknowledgment (DAA).

PARAMETERS:
RETURN_CODE : control_out
ldt_socket_ptr : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_FAILURE    ldt_RcvDaa failed to complete
LPS_SUCCESS    ldt_RcvDaa completed successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
1. DAA_TIMEOUT is a #define constant.
2. This design is based on the ECS/LPS ICD dated July 1995. The current review
   copy has changes that will impact the DAA processing.

End Unit ldt_RcvDaa
*****************************************************************
```

```
UNIT NAME:
  Receive DDN

PURPOSE:
  Establish server session for receiving DDN's, receive DDN's, process
  DDN's, send DDA's

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LDT_NORMAL_SHUTDOWN
               Successful shutdown of DDN server as directed by MACS
LDT_FAILED_TO_CREATE_SERVER
               Could not establish server socket
LDT_DB_CONNECTION_FAILED
               Could not connect to database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_RcvDDN
*****************************************************************
```

```
UNIT NAME:  ldt_read_socket

PURPOSE:
  Read from socket.

PARAMETERS:
ldt_rsckBufArg : data_inout
ldt_rsckSockFdArg : data_in
ldt_rsckMsgLenArg : data_out
ldt_rsckMsgTypeArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type =

Value          Description
-------------- ------------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                        Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
****************************************************************
```

```
UNIT NAME: ldt_readdan

PURPOSE: To read a dan.

PARAMETERS:
RETURN_CODE : control_out
ldt_DAN_seqno : data_in
ldt_sdDANArg : data_out

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ----------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
  1. This is a reuse unit, its PDL is not supplied at this time.

End Unit "Name"
***************************************************************
```

```
UNIT NAME:
  ldt_RetainFiles

PURPOSE:
  Mark for retention the file set identified by input contact seq ID
  and file version ID.

PARAMETERS:
ldt_rtfFileVersionIdArg : data_in
RETURN_CODE : control_out
ldt_rtfContactSeqIdArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_FAILURE    Error accessing database
LPS_SUCCESS    No error accessing database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_RetainFiles
****************************************************************
```

```
UNIT NAME:
  ldt_RsndSuspDANs

PURPOSE:
  Identify the contact sequence ID/file version ID pairs in the database
  for which the associated DAN is suspended and call ldt_SendDAN to send
  DAN's.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_FAILURE    Resend DANs failed to complete.
LPS_SUCCESS    Resend DANs completed.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NONE

NOTES:

NONE

End Unit ldt_RsndSuspDANs
****************************************************************
```

```
UNIT NAME:
  ldt_SendDAN

PURPOSE:
  Accepts requests from the MACS to send or resend Data Availability
  Notifications (DAN's).

PARAMETERS:
RETURN_CODE : control_out
ldt_sdFileSetListArg : data_in
ldt_sdResendFlagArg : control_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ---------------------------------------------------------------
LPS_FAILURE    Fatal error(s) detected during processing.
LPS_SUCCESS    No fatal error(s) during processing.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
---------------- ------------------ --- ----------------------------------------
LDT_File_Set_Info          O  Database table containing file set information.

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
  1.  When MACS enables DAN transfer, the database will automatically
      trigger a routine which invokes this function (Send DAN) for each
      contact sequence ID / file version ID pair in the
      LDT_File_Set_Info schema which has DAN_Suspended
      equal to true.

NOTES:
  1. LDT_DDN_TIMEOUT is defined as 8 hrs.

End Unit ldt_SendDAN
*************************************************************
```

```
UNIT NAME:
  ldt_sendDDA

PURPOSE:
  Send a short DDA.

PARAMETERS:
ldt_sddaDDNArg : data_in
ldt_sddaSockFdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ----------------------------------------------------------------
LPS_SUCCESS    Attempt to write DDA to socket was successful.
LPS_FAILURE    Attempt to write DDA to socket was unsuccessful.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

1.  This unit has been reused from DDF Consumer Simulator code.  Some
    of the PDL has been changed to conform to LPS standards.  Some
    has not.

End Unit ldt_sendDDA
*****************************************************************
```

```
UNIT NAME: ldt_shortdaa

PURPOSE: To process the short daa.

PARAMETERS:
RETURN_CODE : control_out
ldt_msg_length : data_in
ldt_bufaddr : data_in

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ----------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
  1. This is a reuse unit, PDL is not supplied at this time.

End Unit "Name"
****************************************************************
```

```
UNIT NAME:
  ldt_socket_response

PURPOSE:
  Process status of last socket read, i.e., decide if socket is empty,
  whether there was a read error, whether it was shut down at the other
  end, whether the read should time out, or whether we got a message.
  If we got a message, call the appropriate routine to process it.

PARAMETERS:
RETURN_CODE : control_out
ldt_srspSockBuffArg : data_in
ldt_srspStatusArg : control_in
ldt_srspProtocolActiveArg : data_in

RETURN VALUE:
Type = RETURN_CODE
Value          Description
-------------- ------------------------------------------------------------
LPS_FAILURE    Read error on socket
                 or LP DAAC closed socket
                 or timeout on socket
                 or error processing message
LPS_SUCCESS    Socket empty or message on socket processed without
                 incident

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ---------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

NOTES:
1.  This unit has been reused from DDF Consumer Simulator code.  Some
    of the PDL has been changed to conform to LPS standards.  Some
    has not.

End Unit ldt_socket_response
*****************************************************************
```

```
UNIT NAME:ldt_start_client_n_clock

PURPOSE: To establish a client connection and send authentication request.

PARAMETERS:
ldt_port_num : data_out
ldt_ecs_host_id : data_out
RETURN_CODE : control_out
ldt_socket_ptr : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    ldt_start_client_n_clock completed
LPS_FAILURE    ldt_start_client_n_clock failed to complete

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                        Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
1. LDT_MAX_ATTEMPTS is currently defined as 3.
2. LDT_SLEEP_TIME is currently defined as 30 ms.
3. Most routines called by this unit are reuse code with minor modification.

End Unit ldt_start_client_n_clock
*****************************************************************
```

```
UNIT NAME: ldt_write_dan_info

PURPOSE: Write the record in the Data_Prod_Info data structure to
         a file indicated by DAN sequence number.

PARAMETERS:
RETURN_CODE : control_out
ldt_DAN_seqno : data_in
ldt_dpiptr : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    ldt_write_dan_info completed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ---------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit ldt_write_dan_info
*****************************************************************
```

```
UNIT NAME:  lps_CaptureIsRunning

PURPOSE:    This unit checks to see if raw data capture is active.

PARAMETERS:
lpsBooleanType : control_out

RETURN VALUE:
Type =  lpsBooleanType

Value          Description
-------------- -------------------------------------------------------------
LPS_TRUE       True if raw data capture is already active
LPS_FALSE      False if no capture is taking place

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ---------------------------------------------
rdc_PIDTempFile           N/A  Temporary file exists if raw data capture is
                               active.  This unit checks to see if this file
                               exists.

EXTERNAL VARIABLES:
None.

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit lps_CaptureIsRunning
****************************************************************
```

```
UNIT NAME:
        lps_db_Commit

PURPOSE:
        This routine commits the database transactions

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    Successfully committed the database transactions
LPS_FAILURE    failed to commit the database transactions

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The caller must have connected to the database server prior to
           this call

NOTES:
        NONE

End Unit lps_db_Commit
*****************************************************************
```

```
UNIT NAME:
        lps_db_Connect

PURPOSE:
        This routine connects a process to the database server.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     Successfully connected to the database server
LPS_FAILURE     Error in connecting to the database server

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
Source           Name                 Type        Use Description
---------------  -------------------- ----------  --- ----------------------------
Unix environment LPS_DB_UID           char *       I  LPS database login UID
Unix environment LPS_DB_PASSWD        char *       I  LPS database login passwd

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        NONE

End Unit lps_db_Connect
*****************************************************************
```

```
UNIT NAME:
        lps_db_Disconnect

PURPOSE:
        This routine disconnects the subsystem from the database server.

PARAMETERS:
lps_commitArg : control_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    Successfully disconnected from the database server
LPS_EINVAL     Invalid augument
LPS_FAILURE    Error in disconnecting from the database server

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        NONE

End Unit lps_db_Disconnect
*****************************************************************
```

```
UNIT NAME:
        lps_db_ErrorMessage

PURPOSE:
        This unit reformats Oracle error message based on database returned
        error code

PARAMETERS:
lps_dbErrorMsgArg : data_out
lps_dbErrorArg : data_in

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        The caller must have connected to the database server before calling
        this routine.

NOTES:
        NONE

End Unit lps_db_ErrorMessage
****************************************************************
```

```
UNIT NAME:
        lps_db_GetLPSConfiguration

PURPOSE:
        This routine retrieves the LPS configuration table from the database
        server.

PARAMETERS:
lps_configurationArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Successfully retrieved LPS_Configuration table from database
LPS_EDBERROR   Failed to retrieve a record due to database error
LPS_ENORECORD  No record found in the database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LPS_Configuration           I  LPS configuration table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The caller has connected to the database server prior to this call

NOTES:
        1. The caller is responsible for allocating a buffer to hold the
                returned LPS_Configuration record

End Unit lps_db_GetLPSConfiguration
*****************************************************************
```

```
UNIT NAME:
        lps_db_GetSubIntvInfo

PURPOSE:
        This routine retrieves the subinterval information table from the
        database server.

PARAMETERS:
lps_subIntvArg : data_out
lps_subIntvIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    Successfully retrieved Sub_Intv table from the database server
LPS_EINVAL     Invalid augument input
LPS_EDBERROR   Failed to retrieve a record due to database error
LPS_ENORECORD  No record found in the database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Sub_Intv                    I  Subinterval table in database

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The caller has connected to the database server prior to this call

NOTES:
        1. The caller is responsible for allocating a buffer to hold the
                returned Sub_Intv record

End Unit lps_db_GetSubIntvInfo
****************************************************************
```

```
UNIT NAME:
        lps_db_InsertFileInfo

PURPOSE:
        Insert a level 0R file information into the LPS database table

PARAMETERS:
lps_fileNameArg : data_in
lps_subIntvSequenceIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CoDE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Successfully insert L0R file
LPS_EINVAL     Invalid augument input
LPS_EDBERROR   Failed to insert file info into database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------------
LPS_File_Info               O  Database table containing L0R file information

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        The caller has connected to the database server prior to this call

NOTES:

End Unit lps_db_InsertFileInfo
****************************************************************
```

```
UNIT NAME:
        lps_db_Rollback

PURPOSE:
        This routine rolls back the database transactions

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    Successfully rolled back the database transactions
LPS_FAILURE    failed to roll back the database transactions

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The caller must have connected to the database server prior to
           this call

NOTES:
        NONE

End Unit lps_db_Rollback
*****************************************************************
```

```
UNIT NAME:
        lps_FIFOClose
PURPOSE:
        This routine detaches the caller from LPS FIFO channel

PARAMETERS:
lps_fifoKeyArg : data_in

RETURN VALUE:
Type = void

Value           Description
--------------  ------------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_fifoKeyIdMap           I  FIFO key and id lookup table resides in LPS FIFO
                                control shared memory segment

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        Assume current LPS IPC mechanism with automatic flow control is system
        V message queue.

NOTES:
        NONE

End Unit lps_FIFOClose
*****************************************************************
```

```
UNIT NAME:
        lps_FIFOCreate
PURPOSE:
        This routine will create a LPS IPC channel

PARAMETERS:
lps_fifoKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------
LPS_SUCCESS     Successfully open FIFO
LPS_ENOSPC      Maximum number of msg queue ids exceeded system limit
LPS_EUNKNOWN    Unknown error code

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_fifoKeyIdMap          I/O FIFO key and id mapping table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        Assume current LPS IPC mechanism with automatic flow control is system
        V message queue.
NOTES:
        1. This routine is used by LPS globals, not for LPS subsystems.
        2. The lps_fifoKeyIdMap table resides in the designated control
           shared memory area. It is created and initialized by
           lps_rsrcAllocFIFO(), and is updated by lps_FIFOCreate(). It is used
           by lps_FIFOOpen(), lps_FIFOSend(), lps_FIFOReceive(), and
           lps_FIFOClose() routines.
           The lps_fifoKeyIdMap table contains fifoChannelId, fifoKey, fifoId
           and the number of parties using the FIFO channel.
        3. Once a fifo channel is created, this routines will update the
           kernel returned unique fifoId in the corresponding lps_fifoKeyIdMap
           table to indicate the fifo channel is active and is ready for other
           users to attach.
        4. This routine uses msgget() system call to create a message queue.
        5. The effective fifo key is computed as follows:

                fifoKey = GID * LPS_FIFOK + lps_fifoKeyArg

           where lPS_FIFOK is a predefined constant (100 < LPS_FIFOK <1999).

End Unit lps_FIFOCreate
*****************************************************************
```

```
UNIT NAME: lps_FIFOOpen

PURPOSE:   This routine will simply attach the caller to the created IPC channel

PARAMETERS:
lps_fifoKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------
LPS_SUCCESS     Successfully opened a FIFO channel
LPS_ENOEXIST    A FIFO channel does not exist
LPS_FAILURE     Failed to open a FIFO channel

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use  Description
------------------------- ---  ------------------------------------------------
lps_fifoKeyIdMap          I/O  FIFO key and id lookup table resides on LPS FIFO
                                control shared memory segment

EXTERNAL VARIABLES:
Source           Name         Type           use Description
---------------  ------------ -----------    --- ------------------------------
lps_fifoKeyIdMap lps_fifoKey  lpsFIFOKeyType  I  LPS FIFO key

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        Assume current LPS IPC mechanism with automatic flow control is system
        V message queue.

NOTES:  1. When a caller calls this routine, the caller will be attached to
           the control fifo shared memory in addition to be attached to the
           FIFO channel.
        2. This routine uses msgget() system call to open a message queue.
        3. The effective fifo key is computed as follows:
                fifoKey = GID * LPS_FIFOK + lps_fifoKeyArg
           where LPS_FIFOK is a predefined constant (100 < LPS_FIFOK <1999).

End Unit lps_FIFOOpen
*****************************************************************
```

```
UNIT NAME:
        lps_FIFOReceive
PURPOSE:
        This routine retrieves data from LPS IPC FIFO channel .

PARAMETERS:
lps_fifoInfoBytesArg : data_in
lps_fifoWaitArg : control_in
lps_fifoKeyArg : data_in
RETURN_CODE : control_out
lps_fifoInfoBytesRcvArg : data_out
lps_fifoInfoArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------
LPS_SUCCESS     Successfully received data from LPS FIFO channel
LPS_EINVAL      Invalid fifo key or fifo info size is 0
LPS_ENOMSG      Queue is empty and non-blocking is set
LPS_EFAULT      Buffer points to an illegal address
LPS_E2BIG       Queue size is greater than buffer size
LPS_EUNKNOWN    Unknown error code

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_fifoKeyIdMap          I/O FIFO key and id lookup table resides on LPS FIFO
                                control shared memory segment

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        Assume current LPS IPC mechanism with automatic flow control is system
        V message queue.
NOTES:
        1. The caller passes the FIFO key to this routine and this routine
           will map out the corresponding resource id for further operation
        2. lps_fifoWaitArg defines the blocking or non-blocking operation and
           contains the following values:
                a. LPS_WAIT: blocking i/o
                b. LPS_NOWAIT: nonblocking i/o
        3. This routine uses msgrcv() system call to receive message from queue

End Unit lps_FIFOReceive
*****************************************************************
```

```
UNIT NAME: lps_FIFORemove

PURPOSE:   This routine removes the FIFO id, and removes the FIFO from the system

PARAMETERS:
lps_fifoKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     Successfully removed LPS FIFO IPC channel
LPS_FAILURE     Failed to remove LPS FIFO channel

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_fifoKeyIdMap           I  FIFO key and id lookup table resides in
                                LPS FIFO control shared memory segment
EXTERNAL VARIABLES:
Source           Name         Type           use Description
---------------  ------------ -----------    --- ------------------------------
lps_fifoKeyIdMap lps_fifoKey  lpsFIFOKeyType  I  LPS FIFO key

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        Assume current LPS IPC mechanism with automatic flow control is system
        V message queue.

NOTES:  1. The message queue operation type is:
                LPS_RMID
        2. This routine will remove the LPS FIFO channel even the channel still
           contains message queues or some parties still connect to the
           channel. However, a message will be logged if at least a party still
           connects to the channel
        3. This routine use msgctl() system call to remove message queue from
           the system

End Unit lps_FIFORemove
*****************************************************************
```

```
UNIT NAME: lps_FIFOSend

PURPOSE:   This routine sends data into LPS IPC FIFO channel .

PARAMETERS:
lps_fifoInfoBytesArg : data_in
lps_fifoWaitArg : control_in
lps_fifoInfoArg : data_in
lps_fifoKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE
Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     Successfully sent data into lps FIFO IPC channel
LPS_EINVAL      Invalid FIFO key or message type used or message size less than
                zero
LPS_EAGAIN      Message cannot send and non-blocking i/o is set
LPS_EFAULT      Buffer points to an illegal address
LPS_EUNKNOWN    Unknown error code

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
lps_fifoKeyIdMap               I  FIFO key and id lookup table resides in
                                    LPS FIFO control shared memory segment
EXTERNAL VARIABLES: NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:  Assume current LPS IPC mechanism with
                 automatic flow control is system V message queue.

NOTES:  1. The caller passes the FIFO key to this routine and this routine
           will map out the corresponding resource id for further operation
        2. fifoWaitArg defines the blocking or non-blocking operation and
           contains the following values:
                a. LPS_WAIT: blocking i/o
                b. LPS_NOWAIT: nonblocking i/o
        3. This routine uses msgsnd() system call to send a message to a queue

End Unit lps_FIFOSend
*****************************************************************
```

```
UNIT NAME: lps_FileNameCreate

PURPOSE:   This routine creates a LPS L0R output filename with a full path based
           on the file type and subinterval id.
PARAMETERS:
lps_fileSubIntvIdArg : data_in
lps_fileTypeArg : data_in
lps_fileNameArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE
Value           Description
--------------  ------------------------------------------------------------
LPS_SUCCESS     A filename with full path was created successfully
LPS_ENOBUF      Error in malloc
LPS_EINVAL      Invalid auguments
LPS_EDBERROR    Oracle database accessment error
LPS_ETIME       Time conversion error

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:
Source           Name                 Type        Use Description
---------------  -------------------- ----------  --- ----------------------------
Unix environment LPS_OUTFILE_PATH    char *       I  Output disk path
                                                       environment
ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
 1. Assume the caller has already connected to the database server
 2. A default output disk path is used if it can't get from the environment

NOTES:  1. The filename format is as follows:

        "L7_" + MF_Start_Time + "_" + File_Type + "_" + File_Version_Number

        2. The subIntv is a local buffer used for retrieving a
           subinterval record information.
        3. The default output disk path is the user's current directory '.'

End Unit lps_FileNameCreate
*****************************************************************
```

```
UNIT NAME: lps_ListAddTail

PURPOSE:   To add a new node to the tail of a list

PARAMETERS:
lps_listArg : data_inout
lps_listEntryArg : data_in

RETURN VALUE:
Type = void
Value          Description
-------------- ------------------------------------------------------------
NONE

NOTES:
        1. This unit is totally reused from PacorII. The actual name in PacorII
           is:
                xxxAddTail()

        2. Assume LPS operation will only need to add a new node to the end of
           a list.

End Unit lps_ListAddTail
*****************************************************************************/
```

```
UNIT NAME:
        lps_ListAlloc

PURPOSE:
        To allocate and initialize a linked list

PARAMETERS:
lps_listArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------------
NONE

NOTES:
        This unit is totally reused from PacorII. The actual name in PacorII
        is:
                xxxAllocList()

End Unit lps_ListAlloc
*****************************************************************************/
```

```
UNIT NAME:
        lps_ListFreeList

PURPOSE:
        To free an entire list

PARAMETERS:
lps_listArg : data_in
lps_listFreeProcArg : data_in
lps_listFreeEntryArg : data_in

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------
NONE

NOTES:
        1. This unit is totally reused from PacorII. The actual name in PacorII
        is:
                xxxFreeList()

        2. This frees all nodes in the list and the list itself. If
           lps_listFreeProcArg is supplied, it is used to free the contents of
           each node. If it is NULL, the caller is responsible for makiing
           sure the contents of each node is freed.

End Unit lps_ListFreeList
*****************************************************************************/
```

```
UNIT NAME:
        lps_ListNextNode

PURPOSE:
        To find the node after a specific node, optionally using a select
        function

PARAMETERS:
lpsAddressType : data_out
lps_listSelectEntryArg : data_in
lps_listSelectProcArg : data_in
lps_listEntryArg : data_in
lps_listArg : data_inout

RETURN VALUE:
Type = void *

Value          Description
-------------- ------------------------------------------------------------
lps_listentry  A data entry

NOTES:
        1. This unit is totally reused from PacorII. The actual name in PacorII
           is:
                xxxNextNode()

End Unit lps_ListNextMode
*****************************************************************************/
```

```
UNIT NAME:
        lps_ListRemHead

PURPOSE:
        To remove a node from the head of a list

PARAMETERS:
lps_listArg : data_inout
lps_listEntryArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------
NONE

NOTES:
        1. This unit is totally reused from PacorII. The actual name in PacorII
           is:
                xxxRemHead()
        2. Since new node is added to the tail for LPS operation, this routine
           provides the FIFO pattern

End Unit lps_ListRemHead
*****************************************************************************/
```

```
UNIT NAME:
        lps_ListRemTail

PURPOSE:
        To remove a node from the tail of a list

PARAMETERS:
lps_listArg : data_inout
lps_listEntryArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------
NONE

NOTES:
        1. This unit is totally reused from PacorII. The actual name in PacorII
        is:
                xxxRemTail()
        2. Since new node is added to the tail for LPS operation, this routine
           provides the FILO pattern

End Unit lps_ListRemTail
*****************************************************************************/
```

```
UNIT NAME:
        lps_LogMessage

PURPOSE:
        To log lps error or status message into lps_Jounal using syslog

PARAMETERS:
lps_logMessageArg : data_in
lps_logLineNumbArg : data_in
lps_logUnitNameArg : data_in
lps_logPriorNumbArg : data_in

RETURN VALUE:
Type = void

Value           Description
--------------- ------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                    Use Description
----------------------- --- ----------------------------------------------
lps_Journal              O  LPS log file

EXTERNAL VARIABLES:
Source           Name                Type        Use Description
---------------  ------------------- ----------  --- ---------------------------
Unix environment LPS_LOG_STDERR      char *       I  Log to stderr flag

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        The syslog file, lps_Journal, must be configured in /etc/syslog.conf

NOTES:
        1. The message format will be as follows:

                date + priority_level + hostname + ":" + GID + "[" +
                source_unit_name + ":"
                + source_line_number + "]" + message

           For example:

        Aug 21 14:02:47 1R:l7xlsrv:2013 [mac_Main:53] Can't open metadata file
        Aug 21 14:02:50 5R:l7xlsrv:1594 [rdp_Main:78] Start L0R processing
        Aug 21 14:03:05 3R:l7xlsrv:5021 [pcd_db_ReportSceneInfo:78] An attempt
        was made to insert the same record twice(1)

        2. If the message writing to the stderr is set in the environment,
           the message will be written into stderr in addition to the
           lps_Journal log file.

End Unit lps_LogMessage
***************************************************************
```

```
UNIT NAME:
        lps_ProcessChildStatus

PURPOSE:
        To convert the wait status to a LPS specific child exit status

PARAMETERS:
lps_procChildStatusArg : data_out
lps_procWaitStatusArg : data_in

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        1. The child exit status is listed below:

                LPS_SUCCESS             Child exit normally
                LPS_FAILURE             Child exit abnormally
                LPS_SUSPENDED           Child was suspended
                LPS_RESUMED             Child was resumed
                LPS_ABORTED             Child was terminated with core dump

End Unit lps_ProcessChildStatus
***************************************************************
```

```
UNIT NAME:
        lps_ProcessInit

PURPOSE:
        To initialize the processing environment for LPS processes

PARAMETERS:
RETURN_CODE : control_out
lps_procDBArg : control_in
lps_procHandleArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    The process environment was initialized successfully
LPS_ESIGNAL    Error in installing the signal handling
LPS_EDBCON     Error in connecting to database server
LPS_EDCRUN     Can't initialize due to data capture is running

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:  The following signals are general to all LPS subsystems. Once a signal
        is delivered and caught by a subsystem, the subsystem needs to terminate
        gracefully. This routine will install the signal actions needed for
        the following signals once they are caught:

        LPS_SIGABT   - abort process
        LPS_SIGBUS   - hardware fault
        LPS_SIGFPE   - arithmetic exception
        LPS_SIGHUP   - hangup
        LPS_SIGILL   - illegal hardware instruction
        LPS_SIGINT   - terminal interrupt control
        LPS_SIGQUIT  - terminal quit control
        LPS_SIGSEGV  - invalid memory reference
        LPS_SIGTERM  - termination by kill()

End Unit lps_ProcessInit
*****************************************************************
```

```
UNIT NAME:
        lps_ProcessStartChild

PURPOSE:
        To fork a child process, exec the requested file or script and pass
        the child with a specified argv vector list

PARAMETERS:
lps_procChildPIDArg : data_out
lps_procChildOptionArg : data_in
lps_procChildNameArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ----------------------------------------------------------------
LPS_SUCCESS     Successfully forked a child process
LPS_EFORK       Error in forking a child
LPS_EEXEC       Error in execing a child

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        Assume the child process uses fd0, fd1 and fd3 for stand in, standard
        out, and standard error

NOTES:
        1. This routine uses fork() system call to create a child process and
           uses system call execv() to load the child process module or script

End Unit lps_ProcessStartChild
*****************************************************************
```

```
UNIT NAME:
        lps_RsrcAlloc
PURPOSE:
        To create LPS interprocess communication resources

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     All LPS IPC resources created successfully
LPS_FAILURE     Failed to create LPS IPC resources

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_fifoKeyIdMap          I/O FIFO key and id mapping table
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        1. The shared memory resources used by LPS are listed as follows:
           Total number of shared resources: LPS_SHMINTOTAL
          a. RDPS_MFPS_Shared_Memory (lps_shm_RDPMFP)
                Size: LPS_SHMSIZE_RDPMFP
                Number of block: LPS_SHMNBLKS_RDPMFP
                Block size: LPS_SHMBLKSIZE_RDPMFP
                Shared key: LPS_SHMKEY_RDPMFP

          b. MFPS_IDPS_Shared_Memory (lps_shm_MFPIDP)
                Size: LPS_SHMSIZE_MFPIDP
                Number of block: LPS_SHMNBLKS_MFPIDP
                Block size: LPS_SHMBLKSIZE_MFPIDP
                Shared key: LPS_SHMKEY_MFPIDP

          c. MFPS_PCDS_Shared_Memory (lps_shm_MFPPCD)
                Size: LPS_SHMSIZE_MFPPCD
                Number of block: LPS_SHMNBLKS_MFPPCD
                Block size: LPS_SHMBLKSIZE_MFPPCD
                Shared key: LPS_SHMKEY_MFPPCD

        2. The FIFO channels used by LPS are listed as follows:
           Total number of FIFO channels: LPS_FIFOINTOTAL
           a. PCDS_IDPS_FIFO
                FIFO key: LPS_FIFOKEY_IDPPCD

           b. IDPS_CHILD_FIFO
                FIFO key: LPS_FIFOKEY_IDPCHLD
```

```
        3. The FIFO control shared memory segment used by LPS globals
                Size: LPS_FIFOINTOTAL * sizeof(lps_fifoKeyIdMap)
                Shared key: LPS_SHMKEY_FIFO

        4. The shared resource control shared memory segment used by LPS
           globals:
                Size: LPS_SHMINTOTAL * [sizeof(lps_shmKeyIdMap)
                        + sizeof(lps_shmList)]
                Shared key: LPS_SHMKEY_SHM

        All parameters listed above will be defined in MACRO under
        #define.

End Unit lps_RsrcAlloc
*****************************************************************
```

```
UNIT NAME:
        lps_RsrcAllocFIFO
PURPOSE:
        To create LPS FIFO channels

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     All LPS FIFO channels created successfully
LPS_FAILURE     Failed to create LPS FIFO channels

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_fifoKeyIdMap          I/O LPS FIFO key and id lookup table resides in LPS
                              FIFO control shared memory segment

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        The lps_fifoKeyIdMap table contains fifoChannelId, fifoKey, fifoId
        and the number of parties using the FIFO channel.

End Unit lps_RsrcAllocFIFO
*****************************************************************
```

```
UNIT NAME:
        lps_RsrcAllocShm
PURPOSE:
        To create LPS shared memory segments

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     All LPS shared memory segments created successfully
LPS_FAILURE     Failed to create LPS shared memory segments

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
Source           Name          Type           use Description
---------------  ------------  -----------    --- ------------------------------
lps_shmCntrlInfo lps_shmSegPtr lpsAddressType  I  Shared memory segment addrs

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        1. The shared resource control table (lps_shmCntrlInfo) resides on the
           shared resource shared memory segment. It contains the shared
           resource control header and multiple shared resource information
           tables. Each shared resource information table contains the shared
           memory segment header information, the free block list, and the
           active block list.

End Unit lps_RsrcAllocShm
****************************************************************
```

```
UNIT NAME:
        lps_RsrcDealloc
PURPOSE:
        To deallocate LPS interprocess communication resources

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     All LPS shared memory resources deallocated successfully
LPS_PARTIAL     Part of shared memory deallocated successfully
LPS_FAILURE     Failed to deallocate any LPS shared memory resource

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
Source           Name          Type           use Description
---------------  ------------  -----------    --- ------------------------------
lps_shmCntrlInfo lps_shmSegPtr lpsAddressType  I  Shared memory segment addrs

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        1. Refer to the NOTES in lps_RsrcAlloc()
        2. In case an IPC resource can't be deallocated, this routine will
           report the error message and continue to process the next IPC
           resource until all the resources are processed.
        3. Refer to lps_ShmAllocShm for the shared resource control table
           structure definition

End Unit lps_RsrcDealloc
*****************************************************************
```

```
UNIT NAME: lps_ShmAddListTail

PURPOSE:   To add a node into the tail of a list

PARAMETERS:
lps_shmRsrcIdArg : data_in
lps_shmBlkTypeArg : data_in
lps_shmBlkArg : data_in

RETURN VALUE:
Type = void
Value           Description
--------------  ------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
lps_shmCntrlInfo           I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE
NOTES:
        1. The LPS free block list and active block list reside
           in the shared resouce information table. Each list contains the
           list header, and a number of blocks. The list header contains the
           following information:

                TotalNumberOfAvailableFree(Active)Blocks

        2. The individual node contains the following information:

                Blockaddress

        3. If the BlockAddress is null, the block is inactive.

End Unit lps_ShmAddListTail
*****************************************************************
```

```
UNIT NAME:
        lps_ShmClose
PURPOSE:
        To detach from a shared memory resource

PARAMETERS:
lps_shmKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     Detach from a shared memory resource successfully
LPS_ENOEXIST    A shared memory resource does not exist
LPS_FAILURE     Failed to detach a shared memory segment

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        1. This routine will detach all shared memory segments associated with
           the assigned shared memory key
        2. Refer to lps_RsrcAllocShm for the shared resource information table
           structure
        3. Refer to lps_RsrcAllocShm for the shared segment information table
           structure

End Unit lps_ShmClose
****************************************************************
```

```
UNIT NAME:
        lps_ShmComputeSize
PURPOSE:
        To compute total number of shared memory segments needed for a shared
        resource and the partial segment size used

PARAMETERS:
RETURN_CODE : control_out
lps_shmPartSegSizeArg : data_out
lps_shmNumbSegArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ----------------------------------------------------------------
LPS_SUCCESS     Successfully compute the shared memory segments
LPS_EINVAL      Invalid input parameter used for size calculation
LPS_ENOSPC      Number of segments exceeds the system-imposed limit

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        1. This routine calculates the total number of segments needed based
           on the combination of POSIX imposed segment size limit and the
           maximum number of segment size with contiguous blocks in the
           segment.
        2. If the POSIX imposed shared memory segment size is too small, this
           unit will use SGI specific segment size for implementation.

End Unit lps_ShmComputeSize
*****************************************************************
```

```
UNIT NAME:
        lps_ShmCreate
PURPOSE:
        To create the shared memory resource

PARAMETERS:
lps_shmRsrcIdArg : data_in
lps_shmRsrcInfoArg : data_inout
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------
LPS_SUCCESS     A shared memory resource is created successfully
LPS_ENOBUF      Can't not allocate shared memory resource
LPS_ENOSPC      Maximum number of shared memory ids exceeded system limit
LPS_EUNKNOWN    Unknown error code
LPS_EINVAL      Invalid shared memory size requested
LPS_EMFILE      Number of shared memory segments attached exceeds system limit

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
lps_shmCntrlInfo           I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        1. The total size of a shared resource is the product of
           shared memory block size in bytes and total number of shared memory
           blocks needed. If the size exceeds the POSIX defied shared memory
           segment limit, more than one segments will be created.
        2. The effective shared resource key is computed as follows:
                shmRsrcKey = GID * LPS_SHMK + lps_shmKey + lps_shmSegId - 1
           where LPS_SHMK is a predefined constant (100 < LPS_SHMK < 1999).
        3. The shared resource control table (lps_shmCntrlInfo) resides on the
           shared resource shared memory segment. It contains the shared
           resource control header and multiple shared resource information
           tables. Each shared resource information table contains the shared
           memory segment header information, the free block list, and the
           active block list.
        4. Refer to lps_RsrcAllocShm for the shared resource control table
           structure
        5. All created shared memory segments will be chopped into blocks
           based on the block size and chained into the free block pool
        6. The actvie block list will occupy the same size(space) as free
           block list in the shared resource control shared memory segment

End Unit lps_ShmCreate
****************************************************************
```

```
UNIT NAME:
        lps_ShmCreateSemaphore
PURPOSE:
        To create a semaphore set

PARAMETERS:
lps_shmRsrcIdArg : data_in
lps_shmNumbFBlkArg : data_in
lps_shmSemIdArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------
LPS_SUCCESS     A semaphore set was created successfully
LPS_FAILURE     Failed to create a semaphore

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        1. A semaphore set is created to control the active block and the free
           block lists for a shared memory resource. Each semaphore set
           contains three semaphores. One (active semphore) is used to control
           the active block list, the other (free semaphore) is used to
           control the free block list. The third one is used as a lock variable
           to avoid any race conditions during the creation, initialization,
           and closing of the semaphore set.

        2. The effective semaphore key is computed as follows:
                semKey = GID * LPS_SEMK + lps_shmKey
           where LPS_SEMK is a predefined constant (100 < LPS_SEMK < 1999)
        3. Refer to lps_RsrcAllocShm for the SemaphoreOperationArray table
           structure and the shared resource information table structure.

End Unit lps_ShmCreateSemaphore
****************************************************************
```

```
UNIT NAME:
        lps_ShmGetRdBlk
PURPOSE:
        To provide the caller with a shared memory read block

PARAMETERS:
lps_shmWaitArg : control_in
lps_shmRdBlkArg : data_out
lps_shmKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ---------------------------------------------------------------
LPS_SUCCESS     A read block was retrieved successfully
LPS_ENOEXIST    A shared memory resource does not exist
LPS_EEMPTY      The active block pool is empty

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The read block will be retrieved from the active_linked_list in
           FIFO pattern

NOTES:
        1. lps_shmWaitArg defines the blocking or non-blocking operation and
           contains the following values:
                a. LPS_WAIT: blocking i/o
                b. LPS_NOWAIT: nonblocking i/o

End Unit lps_ShmGetRdBlk
*****************************************************************
```

```
UNIT NAME:
        lps_ShmGetWrBlk
PURPOSE:
        To provide the caller with a shared memory write block
PARAMETERS:
lps_shmWaitArg : control_in
lps_shmWrBlkArg : data_out
lps_shmKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ---------------------------------------------------------------
LPS_SUCCESS     A write block was retrieved successfully
LPS_EEMPTY      The free block pool is empty
LPS_ENOEXIST    A shared memory resource does not exist

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The write block will be retrieved from the free_linked_list in
           FILO pattern

NOTES:
        1. The effective shared resource key is computed as follows:
                shmRsrcKey = lps_shmGpidArg * K + lps_shmKeyArg
           where K is a predefined constant (100 < K <1999).
        2. lps_shmWaitArg defines the blocking or non-blocking operation and
           contains the following values:
                a. LPS_WAIT: blocking i/o
                b. LPS_NOWAIT: nonblocking i/o

End Unit lps_ShmGetWrBlk
****************************************************************
```

```
UNIT NAME:
        lps_ShmOpen
PURPOSE:
        To attach the caller to a shared memory resource
PARAMETERS:
lps_shmKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE
Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     A shared memory resource was attached successfully
LPS_ENOEXIST    A shared memory resource does not exist
LPS_EUNKNOWN    Unknown error code
LPS_EMFILE      Number of shared memory segments attached exceeds system limit
LPS_ESEMATTCH   Semaphore reference failed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The shared memory resource must have been created prior this call

NOTES:
        1. The effective shared resource key is computed as follows:
                shmRsrcKey = GID * LPS_SHMK + lps_shmKeyArg
           where LPS_SHMK is a predefined constant (100 < LPS_SHMK <1999).
        2. This routine will attach all shared memory segments associated with
                assigned shared memory key, attach the semaphore
        3. Refer to lps_RsrcAllocShm for the shared resource information table
                structure
        4. Refer to lps_RsrcAllocShm for the shared segment information table
                structure

End Unit lps_ShmOpen
*****************************************************************
```

```
UNIT NAME:
        lps_ShmOpenSemaphore
PURPOSE:
        To attach the caller to a semaphore set

PARAMETERS:
lps_shmKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     A semaphore was attached successfully
LPS_FAILURE     Semaphore reference failed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The semaphore set must have been created prior this call

NOTES:
        1. The effective semaphore key is computed as follows:
                semKey = GID * LPS_SEMK + lps_shmKey
           where LPS_SEMK is a predefined constant (100 < LPS_SEMK < 1999)
        2. This routine will attach the caller to the existing semaphore

End Unit lps_ShmOpenSemaphore
*****************************************************************
```

```
UNIT NAME:
        lps_ShmPutRdBlk
PURPOSE:
        To release a read block and place the read block into the
        free block pool

PARAMETERS:
lps_shmRdBlkArg : data_in
lps_shmKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ----------------------------------------------------------------
LPS_SUCCESS     A read block was placed into free block pool successfully
LPS_ENOEXIST    A shared memory resource does not exist
LPS_EINVAL      An invalid block address used

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
Source           Name             Type           use Description
---------------  ---------------  -----------    --- -------------------------
lps_shmCntrlInfo lps_shmCntrlInfo lpsAddressType  I  Shared memory segment
                                                       address

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The read block will be added to the free_block_list in
           FIFO pattern

NOTES:
        None

End Unit lps_ShmPutRdBlk
****************************************************************
```

```
UNIT NAME:
        lps_ShmPutWrBlk
PURPOSE:
        To release a write block and place the write block into the active
        block pool

PARAMETERS:
lps_shmWrBlkArg : data_in
lps_shmKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     A write block was placed into active block pool successfully
LPS_ENOEXIST    A shared memory resource does not exist
LPS_EINVAL      An invalid block address used

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
Source           Name             Type           use Description
---------------  ---------------  -----------    --- -------------------------
lps_shmCntrlInfo lps_shmCntrlInfo lpsAddressType  I  Shared memory segment
                                                       address

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The write block will be added to the active_block_list in
           FIFO pattern

NOTES:
        None

End Unit lps_ShmPutWrBlk
*****************************************************************
```

```
UNIT NAME:
        lps_ShmRemListHead
PURPOSE:
        To remove a list node from the head of active block list
PARAMETERS:
lps_shmRsrcIdArg : data_in
lps_shmBlkArg : data_out

RETURN VALUE:
Type = void
Value           Description
--------------  ------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                        Use Description
--------------------------- --- ------------------------------------------------
lps_shmCntrlInfo            I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE
NOTES:
        1. The LPS free block list and active block list reside
           in the shared resouce information table. Each list contains the
           list header, and a number of blocks. The list header contains the
           following information:

                TotalNumberOfAvailableFree(Active)Blocks

        2. The individual node contains the following information:

                BlockAddress

        3. If the BlockAddress is null, the block is inactive

End Unit lps_ShmRemListHead
*****************************************************************
```

```
UNIT NAME:
        lps_ShmRemListTail
PURPOSE:
        To remove a list node from the tail of free block list
PARAMETERS:
lps_shmRsrcIdArg : data_in
lps_shmBlkArg : data_out

RETURN VALUE:
Type = void
Value           Description
--------------  ------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
lps_shmCntrlInfo           I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:  NONE

NOTES:
        1. The LPS free block list and active block list reside
           in the shared resouce information table. Each list contains the
           list header, and a number of blocks. The list header contains the
           following information:

                TotalNumberOfAvailableFree(Active)Blocks

        2. The individual node contains the following information:

                BlockAddress

        3. If the BlockAddress is null, the block is inactive

End Unit lps_ShmRemListTail
*****************************************************************
```

```
UNIT NAME:
        lps_ShmRemove
PURPOSE:
        To remove a shared memory resource from the system

PARAMETERS:
lps_shmKeyArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------  ------------------------------------------------------------------
LPS_SUCCESS     A shared memory resource was removed successfully
LPS_ENOEXIST    A shared memory resource did not exist

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
lps_shmCntrlInfo          I/O Shared resource control information table

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        None

NOTES:
        1. This routine will remove all shared memory segments associated with
                the assigned shared memory key
        2. Refer to lps_RsrcAllocShm for the shared resource control table
           structure

End Unit lps_ShmRemove
*****************************************************************
```

```
UNIT NAME:
        lps_TimeAdd

PURPOSE:
        To add a time duration to a time value

PARAMETERS:
lps_timeOffsetArg : data_in
lps_timeArg : data_inout

RETURN VALUE:
Type = void

Value          Description
-------------- ----------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:

End Unit lps_TimeAdd
*************************************************************
```

```
UNIT NAME:
        lps_TimeCompare

PURPOSE:
        To compare two time values and return an integer greater than, equal
        to or less than 0 based on first time value is greater than, equal
        to, or less than second time value.

PARAMETERS:
lps_time2Arg : data_in
lps_time1Arg : data_in
lpsUInt16Type : data_out

RETURN VALUE:
Type = lpsUInt16Type

Value          Description
-------------- ------------------------------------------------------------------
> 0            Time1 is greater than time2
= 0            Time1 is equal to time2
< 0            Time1 is less than time2

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        Assume the two string times are the same structure format

NOTES:
        This routine compares the two time values based on the number of
        characters in the time value.

End Unit lps_TimeCompare
*****************************************************************
```

```
UNIT NAME:
        lps_TimeDiff

PURPOSE:
        To compare two time values and return the difference and comparison
        status

PARAMETERS:
RETURN_CODE : control_out
lps_time2Arg : data_in
lps_time1Arg : data_in
lps_timeOffsetArg : data_out

RETURN VALUE:
Type = lpsUInt16Type

Value          Description
-------------- ------------------------------------------------------------
LPS_GT         Time1 is greater than time2
lPS_EQ         Time1 is equal to time2
LPS_LT         Time1 is less than time2

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        Assume the two times are the same structure format or same string
        format

NOTES:
        NONE

End Unit lps_TimeDiff
*************************************************************
```

```
UNIT NAME:
        lps_TimeDivide

PURPOSE:
        To calculate the total number of time units for a time duration

PARAMETERS:
RETURN_CODE : control_out
lps_time2Arg : data_in
lps_time1Arg : data_in
lps_timeNumbUnitArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Time division successfully
LPS_EFAULT     Time divided by zero or negative number

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        NONE

End Unit lps_TimeDivide
****************************************************************
```

```
UNIT NAME:
        lps_TimeGetCurrentTime

PURPOSE:
        Return system time in yyyy-mm-dd-hh-mm-ss format.

PARAMETERS:
lps_timeCurrentTimeArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        1. tm struct is defined in time.h.
        2. UTC stands for Coordinated Universal Time, used as the UNIX system
time.
        3. The actual output time format may vary from current yyyy-mm-dd-hh-mm-
ss
           format later on.

End Unit lps_GetCurrentTime
*************************************************************
```

```
UNIT NAME:
        lps_TimeString2Struct

PURPOSE:
        To convert a time string into a time structure

PARAMETERS:
lps_timeStructFormatArg : data_in
lps_timeStringArg : data_in
lps_timeArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        NONE

End Unit lps_TimeString2Structure
*****************************************************************
```

```
UNIT NAME:
        lps_TimeStruct2String

PURPOSE:
        To convert an input time structure into a time string
        based on the input time format

PARAMETERS:
lps_timeStructFormatArg : data_in
lps_timeArg : data_in
lps_timeStringArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
        NONE

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        NONE

NOTES:
        NONE

End Unit lps_TimeStruct2String
*****************************************************************
```

```
UNIT NAME: mac_ACAcceptSignal

PURPOSE:

Handle auto capture control signals

PARAMETERS:

RETURN VALUE:
Type = void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
mac_Globals     mac_acSignalNumberReceived
                                    mac_acSignalType
                                                    O   Set to signal number
                                                        of received signal
ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

The function assumes signal handler has been installed to get the signal number
Cf. sigaction(2) for details

NOTES:

End Unit mac_ACAcceptSignal
***************************************************************
```

```
UNIT NAME: mac_ACSleep

PURPOSE:

  Sleep for specified time or until receipt of auto capture control signals

PARAMETERS:
mac_acSleepTimeArg : data_in

RETURN VALUE:
Type = void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

1.  The 3 step operation of unblocking signals, pausing, and restoring the
original
    signal mask can be handled by the POSIX sigsuspend(2) function

End Unit mac_ACSleep
****************************************************************
```

UNIT NAME: mac_AutoCapture

PURPOSE:

Main function for automatic, scheduled raw data capture program

PARAMETERS:

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Successful execution
LPS_FAILURE	Execution aborted because attempt to register as raw capture program failed.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mac_Globals	mac_acSignalNumberReceived	mac_acSignalType	I/O	MAC_AC_NOSIGNAL = no signal received or signal number receive

NOTES:
1. Receipt of MAC_AC_SCHED_UPDATE_SIG (indicating the contact schedule has been modified) causes the process to fetch the next scheduled capture time. Receipt of LPS_AC_TERM_SIG causes the process to exit.
2. Signals are blocked except when in a sleep wait (in mac_ACSleep) for the next scheduled capture event. Any signal sent at another time is handled then.
3. Signal handler should be installed so that it receives the signal number. Cf. sigaction(2) for a description of how to do this.
4. This is a daemon process, so it detaches from its controlling terminal and sets itself up as process group leader in a new process group.

End Unit mac_AutoCapture
****

UNIT NAME: mac_db_ACGetNextCaptureInfo

PURPOSE:

Fetch info on next contact from contact schedule in LPS database

PARAMETERS:
RETURN_CODE : control_out
mac_acNextCaptureInfoArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Next capture event was retrieved
LPS_FAILURE	No capture event retrieved (due to receipt of termination signal)

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
Contact_Schedules	I	Database table containing contact schedules

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mac_Globals	mac_acSignalNumberReceived	mac_acSignalType	I	MAC_AC_NOSIGNAL = no signal received or signal number received

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
1. All signals are ignored here. The value of mac_acSignalNumberReceived may change only when the process is sleeping
2. Control remains in process until either info is retrieved or a termination signal is received

End Unit mac_db_ACGetNextCaptureInfo
****

```
UNIT NAME: mac_db_ACRegisterCapture

PURPOSE:

Register process as the automatic capture process

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    The process was registered as the automatic capture process
LPS_FAILURE    The process was not registered as the automatic capture process

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_db_ACRegisterCapture
*************************************************************
```

```
UNIT NAME:  mac_db_GetRDCFilename

PURPOSE: To retrieve a raw data capture filename from the LPS data base.

PARAMETERS:
mac_fileVerNumArg : data_in
mac_contSeqIdArg : data_in
mac_rdcFilenameArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Able to retrieve filename from database
LPS_FAILURE    Unable to retrieve filename from database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -----------------------------------------------
RDC_Acct                   I  Raw data capture accounting table

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) None

NOTES:
None

End Unit "mac_db_GetRDCFilename"
*****************************************************************
```

```
UNIT NAME:  mac_db_IASUpdateParm

PURPOSE:    Update a single IAS parameter value in the LPS database

PARAMETERS:
mac_iasParmValueArg : data_in
mac_iasParmNameArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE
Value                   Description
--------------          -------------------------------------------------------
LPS_SUCCESS             Parameter was successfully updated
LPS_FAILURE             Parameter was not successfully update
MAC_IAS_ALLIASPARMS     Parameter was successfully updated; all IAS parameters
have
                        been updated.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ---------------------------------------------
Valid_MFP_Parms           O   LPS DB Table containing sensor alignment info
Valid_Scene_Parms         O   LPS DB Table containing WRS scene ID parameters

EXTERNAL VARIABLES: None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

  1. Assumes that Oracle handles conversion from ASCII to appropriate data type
  2. Assumes that triggers perform validation.

NOTES:
  1.  The function definition is purposely general pending complete
  LPS/IAS I/F definition.

  2.  The function keeps local  track of attributes updated and indicates when
  all parameters are updated in its return value.

End Unit mac_db_IASUpdateParm
*****************************************************************
```

```
UNIT NAME:  mac_db_Initialize

PURPOSE:  Initailize database tables

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Successfully initalized database tables
LPS_FAILURE    Execution aborted beacause initializing database tables failed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Process_Id                  I  Database table containing LPS process information

EXTERNAL VARIABLES:
NONE

 ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

 NOTES:

End Unit mac_db_Initialize
*****************************************************************
```

```
UNIT NAME:  mac_db_MetaDataGetAScene

PURPOSE:  Get the metadata scene information for the indicated scene and places
          the information into a data structure.

PARAMETERS:
mac_pathNomArg : data_in
mac_rowNomArg : data_in
mac_sceneCtrTimeArg : data_in
mac_subIntvSeqIdArg : data_in
RETURN_CODE : control_out
mac_sceneArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS
LPS_FAILURE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
PCD_Scene_Info              I  Database table containing aggregate scene
accounting
                                information from the PCDS on a scene basis
IDP_Acct                    I  Database table containing aggregate accounting
                                information from the IDPS on a subinterval basis
Valid_WRS_Parms             I  Database table containing information for each
                                Worldwide Reference System scene

EXTERNAL VARIABLES:
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
NONE

 NOTES:

End Unit mac_db_MetaDataGetAScene
*************************************************************
```

```
UNIT NAME: mac_db_MetaDataGetBandsPresent

PURPOSE: Get the first bands present byte of a subinterval
         (i.e., subinterval stop time >= its time >= subinterval start time)

PARAMETERS:
mac_formatArg : data_in
mac_subIntvStopTimeArg : data_in
mac_subIntvStartTimeArg : data_in
mac_fileVerNumArg : data_in
mac_contSeqIdArg : data_in
mac_bandsPresentArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS
LPS_FAILURE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Bands_Present               I  Database table containing bands present information

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_db_MetaDataGetBandsPresent
****************************************************************
```

```
UNIT NAME:  mac_db_MetaDataGetSubIntv

PURPOSE:  Get the subinterval metadata information from the LPS database and
place
          the information into the data structures.

PARAMETERS:
mac_fileVerNumArg : data_in
mac_contSeqIdArg : data_in
mac_subIntvSeqIdArg : data_in
RETURN_CODE : control_out
mac_subIntvArg : data_inout
mac_fileNameArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS
LPS_FAILURE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
MFP_Acct                    I  Database table containing aggregate accounting
                               information from the MFPS on a subinterval basis
PCD_Acct                    I  Database table containing aggregate accounting
                               information from the PSDS on a subinterval basis
RDC_Acct                    I  Database table containing aggregate accounting
                               information from the RDCS on a subinterval basis
IDP_Acct                    I  Database table containing aggregate accounting
                               information from the IDPS on a subinterval basis
LPS_File_Info               I  Database table containing LPS file information
                               from the LDTS on a subinterval basis

EXTERNAL VARIABLES:
NONE

End Unit mac_db_MetaDataGetSubIntv PDL:
***************************************************************
```

UNIT NAME: mac_db_RegL0RPID

PURPOSE: To register a Level 0R parent process identifier into the data base.

PARAMETERS:
mac_contSeqIdArg : data_in
mac_fileVerNumArg : data_in
mac_pidArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Able to insert process identifier into data base
LPS_FAILURE	Unable to insert process identifier into data base

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
Process_Id	O	LPS process id table

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
None				

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
None

End Unit "mac_db_RegL0RPID"
****

UNIT NAME: mac_db_RollbackL0R

PURPOSE: To rollback the database to the state prior to the L0R processing of this contact.

PARAMETERS:
mac_contSeqIdArg : data_in
RETURN_CODE : control_out
mac_fileVerNumArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Able to rollback the database to prior Level 0R status
LPS_FAILURE	Unable to rollback the database to prior Level 0R status

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
Bands_Present	I	Bands present database table
IDP_Acct	I	IDPS accounting database table
LPS_File_Info	I	LPS file information database table
MFP_Acct	I	MFP accouting database table
PCD_Acct	I	PCD accounting database table
PCD_Scene_Acct	I	PCD scene accounting database table
Process_Id	I	Process identifier database table
RDP_Acct	I	RDP accounting database table
Sub_Intv	I	Subinterval database table

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
None				

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) If a deletion error is encountered, deletion of further records halts. It is expected that the operator or the DBA will have to rectify the database.

End Unit mac_db_RollbackL0R
****

UNIT NAME: mac_db_UnregL0RPID

PURPOSE: To delete a parent process identifier from the data base.

PARAMETERS:
mac_pidArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Able to delete a parent process identifier from the database.
LPS_FAILURE	Unable to delete a parent process identifier from the database.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
Process_Id	O	LPS process id table

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
None				

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit mac_db_UnregL0RPID
****

UNIT NAME: mac_ExamLPSJournal

PURPOSE: Open a window and display messages from LPS journal

PARAMETERS:

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Opened a window and successfully displayed the messages from the LPS journal
LPS_FAILURE	Could not successfully display the messages

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
LPS_Journal	I	LPS Log Journal

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
Relies on the COTS product for scrolling, resizing, an exiting the window

End Unit mac_ExamLPSJournal
****

```
UNIT NAME:  mac_FilterMsg

PURPOSE:  Filter messages whose numerical priorities are less than the operator
provided
          priority from the LPS journal and write them to the monitor window

PARAMETERS:
RETURN_CODE : control_out
mac_opsSeverityArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Successfully fitered and dislayed messages from the LPS journal to
the window
LPS_FAILURE    Could not filter and display the messages to the window

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                    Use  Description
----------------------- ---- ----------------------------------------
LPS_Journal              I   LPS Log journal

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
   Messages are filtered whose numerical priority is less than the
   operator provided priority

End Unit mac_FilterMsg
****************************************************************
```

```
UNIT NAME: mac_GenQAReport

PURPOSE:  Generate the LPS Quality and Accounting Report

PARAMETERS:
RETURN_CODE : control_out
mac_rptContactSeqIdArg : data_in
mac_rptFileVerNumArg : data_in

RETURN VALUE:
Type = RETURN_CODE
Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS
LPS_FAILURE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use    Description
-----------------------  ---    ------------------------------------------------
-
IDP_Acct                   I     Database table containing aggregate accounting
                                 information from the IDPS on a subinterval basis
LPS_Configuration          I     Database table containing LPS configuration
information
MFP_Acct                   I     Database table containing aggregate accounting
                                 information from the MFPS on a subinterval basis
RDC_Acct                   I     Database table containing raw data capture
information
                                 from the RDCS on a contact basis
RDP_Acct                   I     Database table containing aggregate accounting
                                 information from the RDPS on a contact basis
Sub_intv                   I     Database table containing subinterval information
Valid_CCSDS_Parms          I     Database table containing valid CCSCS parameter
list

EXTERNAL VARIABLES:  NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. The Oracle Reports is used to generate the LPS QA report defined in this
specification
2. Connection/Disconnection to/from the database is through the Oracle Reports

End Unit mac_GenQAReport PDL:
*************************************************************
```

UNIT NAME: mac_IngestIASParms

PURPOSE:

Ingest IAS parameter file into LPS database

PARAMETERS:
mac_ingestIASFileNameArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	IAS parameter file information is successfully entered into the database
LPS_FAILURE	IAS parameter file information is not successfully entered into the database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
IAS_Parameter_File	I	IAS Parameter File in PVL HDF format

EXTERNAL VARIABLES:

None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

1. The IAS parameter file is an HDF format file containing PVL

NOTES:

1. Many details of this function's behavior are TBD pending IAS/LPS I/F definition.

End Unit mac_IngestIASparms
****

```
UNIT NAME:  mac_LPS

PURPOSE:  Start up the LPS system

PARAMETERS:

RETURN VALUE:
Void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) The LPS database is up and running before mac_LPS is started.
   mac_AutoCapture and ldt_RcvDDN both enter their own process id into the
   Process_Id table.

NOTES:

End Unit mac_LPS
****************************************************************
```

```
UNIT NAME: mac_MainAbortL0R

PURPOSE: To abort the active L0R subsystems

PARAMETERS:

RETURN VALUE:
Type = void

Value         Description
------------- ------------------------------------------------------------------
None

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
mac_L0RGlobal   mac_processArray     lpsInt32Type []  I  Process array

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit mac_AbortL0R
*****************************************************************
```

UNIT NAME: mac_MainActivateChildProcesses

PURPOSE: To activate the L0R child processes

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Able to activate the L0R child processes
LPS_FAILURE	Unable to activate the L0R child processes

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
None		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mac_L0RGlobal	mac_processArray	lpsInt32Type []	I	Process array
mac_L0RGlobal	mac_statusArray	lpsInt32Type []	I	Status array
mac_L0RGlobal	mac_contSeqId	lpsUInt32Type	I	Contact sequence ID
mac_L0RGlobal	mac_fileVerNum	lpsUInt32Type	I	File version number
mac_L0RGlobal	mac_rdcFilename	lpsStringType	O	The RDC file associated with this contact

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit mac_MainActivateChildProcesses
****

UNIT NAME: mac_MainCleanupL0R

PURPOSE: Cleanup the L0R processing

PARAMETERS:

RETURN VALUE:
Type = void

Value	Description
None	

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
None		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mac_L0RGlobal	mac_dbConnected	lpsBooleanType	O	A flag signifying connection to the database
mac_L0RGlobal	mac_registered	lpsBooleanType	O	A flag signifying registration in the database
mac_L0RGlobal	mac_rsrcsAllocated	lpsBooleanType	O	A flag signifying that the L0r resources have been allocated

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit mac_MainCleanupL0R
****

UNIT NAME: mac_MainInitL0R

PURPOSE:
PARAMETERS:
mac_mainDelRDCFileArg : control_in
mac_mainFileVerNumArg : data_in
mac_mainContSeqIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Successful initialization
LPS_FAILLURE	Unsuccessful initialization

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mac_L0RGlobal	mac_dbConnected	lpsBooleanType	O	A flag signifying connection to the database
mac_L0RGlobal	mac_registered	lpsBooleanType	O	A flag signifying registration in the database
mac_L0RGlobal	mac_rsrcsAllocated	lpsBooleanType	O	A flag signifying that the L0r resources have been allocated
mac_L0RGlobal	mac_contSeqId	lpsUInt32Type	O	The contact sequence ID
mac_L0RGlobal	mac_fileVerNum	lpsUInt32Type	O	The file version number
mac_L0RGlobal	mac_delRDCFile	lpsBooleanType	O	A flag indicating that the RDC file is to be deleted upon completion of L0R processing
mac_L0RGlobal	mac_rdcFilename	lpsStringType	O	The RDC file associated with this contact

End Unit mac_MainInitL0R
****

UNIT NAME: mac_MainRollbackL0R

PURPOSE: To rollback the database and file system to the state prior to the L0R processing of this contact.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Successful in returning database and file system to prior state
LPS_FAILURE	Unsuccessful in returning database and file system to prior state

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
None		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mac_L0RGlobal	mac_contSeqId	lpsUInt32Type	I	Contact sequence ID
mac_L0RGlobal	mac_fileVerNum	lpsUInt32Type	I	File version number
Unix environ	LPS_OUTFILE_PATH	char *	I	Output disk path environment

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) L0R Subsystems are responsible for the deletion of their temp files. All L0R files will be deleted if placed into the correct path identified by an environment variable.

NOTES:
None

End Unit mac_MainRollbackL0R
****

```
UNIT NAME: mac_MainShutdownL0R

PURPOSE: Shutdown the associated L0R processing.

PARAMETERS:

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
None

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:

End Unit mac_MainShutdownL0R
****************************************************************
```

```
UNIT NAME:  mac_MainShutdownLPS

PURPOSE:  Shut down LPS.

PARAMETERS:

RETURN VALUE:
Type = RETURN_CODE

Value           Description
--------------- ------------------------------------------------------------
LPS_SUCCESS     All processes are killed. Exit the form
LPS_FAILURE     Some processes are not kill. Do not exit the form

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                           Use  Description
-----------------------------  ---  -----------
Process_Id                      I   Database table containing LPS process id

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_MainShutdownLPS
*****************************************************************
```

```
UNIT NAME:  mac_MainStartL0R

PURPOSE: Main driver for the LPS L0R processing.

PARAMETERS:
mac_mainDelRDCFileArg : control_in
mac_mainFileVerNumArg : data_in
mac_mainContSeqIdArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    The L0R processing completed successfully
LPS_FAILURE    The L0R processing failed to complete

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) The designer assumes that the process_id and the process name are stored in
   Process_Id table.

NOTES:
1) If the Send DAN process does not complete successfully, the L0R processing
   is not rolled back.  A resend DAN command is expected to be initiated at
   a later time.
2) mac_MainShutdownL0R is setup as a termination signal handler, and is used as
   the shutdown routine.  It performs an exit process which will resullt in
   this process returning a RETURN_CODE to the calling process.  However, this
   routine does not explicitly "RETURN" when it calls the cleanup routine.

End Unit mac_MainStartL0R
*****************************************************************
```

```
UNIT NAME: mac_MainStopL0R

PURPOSE: To halt a particular L0R main process.

PARAMETERS:
mac_mainPidArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    No errors encountered stopping L0R processing
LPS_FAILURE    Encountered errors stopping L0R processing

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit mac_MainStopL0R
*****************************************************************
```

```
UNIT NAME: mac_MainWaitForL0R

PURPOSE: To wait until Level zero processing completes

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    The L0R processes completed successfully
LPS_FAILURE    The L0R processes were aborted prematurely

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
mac_L0RGlobal   mac_processArray     lpsInt32Type []  I  Process array
mac_L0RGlobal   mac_statusArray      lpsInt32Type []  I  Status array

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit mac_MainWaitForL0R
*****************************************************************
```

```
UNIT NAME: mac_MetaDataGen

PURPOSE:  Driver for the generate metadata task.

PARAMETERS:
RETURN_CODE : control_out
mac_metaContactSeqIdArg : data_in
mac_metaFileVerNumArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS
LPS_FAILURE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:
Source              Name                Type            Use Description
------------------  ------------------- --------------- --- ---------------------
--
mac_MetaDataGlobal  mac_metaFileDesc    FILE *          N/A Meta data file
descriptor

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_GenMetadata
****************************************************************
```

```
UNIT NAME:  mac_MetaDataGenFileHeadDesc

PURPOSE:  Get metadata file descriptor information and write them to the meta
data     file

PARAMETERS:
RETURN_CODE : control_out
mac_metaFileCrtTimeArg : data_in
mac_metaFileVerNumArg : data_in
mac_metaFileSrcIdArg : data_in
mac_metaSwVerNumArg : data_in
mac_metaFileNameArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS
LPS_FAILURE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------------
Metadata_File               O  The file where meta data file information is
written

EXTERNAL VARIABLES:
Source              Name                Type      Use Description
------------------  ------------------- --------  --- ----------------------------
----------
mac_MetaDataGlobal  mac_metaFileDesc     FILE *   I   Metadata file descriptor

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_MetaDataGenFileHeadDesc PDL:
*************************************************************
```

```
UNIT NAME:  mac_MetaDataGenScene

PURPOSE:  Generate the scene records of the metadata file for the indicated
          subinterval and write them to the metadata file.

PARAMETERS:
mac_metaSubIntvSeqIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS
LPS_FAILURE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------  --- ------------------------------------------------
---
IDP_Acct                   I   Database table containing IDP accounting
information

EXTERNAL VARIABLES:
Source             Name               Type    Use Description
------------------ ----------------- ------  --- -----------------------
mac_MetaDataGlobal  mac_metaFileDesc FILE *   I  Metadata file descriptor

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
metaSceneStruct         Metadata_Scene_Struct

NOTES:

End Unit mac_MetaDataGenScene PDL:
*****************************************************************
```

```
UNIT NAME:  mac_MetaDataGenSubIntv

PURPOSE:  Get subinterval metadata information and write them to the meta data
          file

PARAMETERS:
RETURN_CODE : control_out
mac_metaFileVerNumArg : data_in
mac_metaContactSeqIdArg : data_in
mac_metaSubIntvSeqIdArg : data_in
mac_metaLpsConfArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------------
LPS_SUCCESS
LPS_FAILURE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:
Source              Name              Type        Use Description
------------------  ----------------  ----------- --- -----------------------
mac_MetaDataGlobal  mac_fileNameDesc  FILE *       I  Meta data file descriptor

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:

End Unit mac_MetaDataGenSubIntv PDL:
*****************************************************************
```

```
UNIT NAME:  mac_MetaDataWriteAScene

PURPOSE:  Write a scene metadata information from the struct into
          the metadata file

PARAMETERS:
mac_metaSceneArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS
LPS_FAILURE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------------
Metadata_File               O   File where the scene information is to be written

EXTERNAL VARIABLES:
Source              Name                Type    Use   Description
------------------ -----------------    ------  ----  ---------------------
mac_MetaDataGlobal  mac_metaFileDesc    FILE *   I    Metadata file descriptor

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_MetaDataWriteAScene PDL:
*****************************************************************
```

```
UNIT NAME:  mac_MetaDataWriteSubIntv

PURPOSE:  Write the subinterval metadata information from the
          structures to the metadata file

PARAMETERS:
RETURN_CODE : control_out
mac_metaFileNameArg : data_in
mac_metaSubIntvArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS
LPS_FAILURE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -------------------------------------------------
Metadata_File              O  The file where metadata information is written

EXTERNAL VARIABLES:
Source              Name              Type      Use Description
------------------  ----------------  -------   --- --------------------------
mac_MetaDataGlobal  mac_metaFileDesc  FILE *    I   Metadata file descriptor

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_MetaDataWriteSubIntv PDL:
*****************************************************************
```

UNIT NAME: mac_OpsMsg

PURPOSE: Open a window and display filtered operations messages

PARAMETERS:
RETURN_CODE : control_out
mac_opsSeverityArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Opened a window and successfully displayed the messages
LPS_FAILURE	Could not successfully display the messages

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

The argument entered by the operator indicates the numerical priority of log messages to be used as the filter priority. Messages whose priority is less than the operator provived priority will be filtered to be displayed.

End Unit mac_OpsMsg
****

```
UNIT NAME: mac_ui_ContSchedule

PURPOSE:  This form allows users to view, insert, modify or delete the contact
schedules table.

PARAMETERS:

RETURN VALUE:  None
Type =  Void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use   Description
-------------------------- ---   ------------------------------------------------
Contact_Schedules           I/O  contact schedules
LPS_Configuration           I    LPS configuration

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_ContSchedule
*****************************************************************
```

```
UNIT NAME:mac_ui_DANTransState

PURPOSE: The form allows the user to change the flag which indicates whether or
not LPS
         DAN transfer is enabled.

PARAMETERS:

RETURN VALUE:
Type = Void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use   Description
-------------------------- ----  ------------------------------------------------
LDT_DAN_Transfer_State     I/O  DAN transfer state

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_DANTransState
****************************************************************
```

```
UNIT NAME: mac_ui_DataRcvSum

PURPOSE: Generate a Data Receive Summary Report for a contact period.
         After the report has been generated, the user will
         be prompted to select to display on the screen and/or output to the
printer.

PARAMETERS:

RETURN VALUE:
Type = Void

Value          Description
-------------- ------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
RDC_Acct                    I  Raw Data Capture accounting table

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_DataRcvSum
*************************************************************
```

```
UNIT NAME:mac_ui_DeleteFileSet

PURPOSE:  This form allows the operator to delete LPS output files of a contact
period
          whose File_Set_Status in the LDT_File_Set_Info table is not LDT_Done

PARAMETERS:

RETURN VALUE:  None
Type = Void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LDT_File_Set_Info           I  Database table containing file set status

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_DeleteFileSet
****************************************************************
```

```
UNIT NAME:mac_ui_ExamLPSJournal

PURPOSE:  The form allows the user to view the whole LPS journal

PARAMETERS:

RETURN VALUE:
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LPS_Configuration           I  Get string id from LPS Configuration table

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_MainExamLPSJournal
****************************************************************
```

```
UNIT NAME:mac_ui_FileTransSum

PURPOSE:  This form allows users to generate file transfer summary report for a
          specified time range.  User must enter the start and stop time. After
report is
          generated, user can select to display on the screen or output to a
printer.

PARAMETERS:

RETURN VALUE:
Type = Void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_FileTransSum
*************************************************************
```

```
UNIT NAME mac_ui_IngestIASParms

PURPOSE: This form askes the operator to enter the file name and allows to invoke
         mac_IngestIASParms.

PARAMETERS:

RETURN VALUE:
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_IngestIASParms
*************************************************************
```

```
UNIT NAME: mac_ui_Level0RQA

PURPOSE:  This form allows user to generate level OR QA report for a specified
          contact period.   This a view only form.

PARAMETERS:

RETURN VALUE:  None
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ----------------------------------------------
RDC_Acct                    I  Raw data caputure accounting

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_Level0RQA
*****************************************************************
```

UNIT NAME: mac_ui_LPSConfig

PURPOSE: This form allows users to view or update LPS Configuration.

PARAMETERS:

RETURN VALUE:
Type = Void

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
LPS_Configuration	I/O	LPS configuration

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
It is assumed that the operator will not change the LPS Configuration during processing.

NOTES:

End Unit mac_ui_LPSConfig
****

UNIT NAME: mac_ui_LPSParms

PURPOSE: The form allows users to view or update parameters for RDPS, MFPS, PCDS or IDPS.

PARAMETERS:

RETURN VALUE:
Type = Void

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
Valid_Band_Parms	I/O	Band Parameters
Valid_CCSDS_Parms	I/O	CCSDS Parameters
Valid_MFP_Parms	I/O	Major Frame Processing Parameters
Valid_PCD_Parms	I/O	Payload Correction Data Parameters
Valid_Scene_Parms	I/O	Scene Parameters

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
It is assumed that the operator will not change the parameters during the processing.

NOTES:

End Unit mac_ui_LPSParms
****

```
UNIT NAME: mac_ui_LPSQA

PURPOSE: This form allows users to generate a report of the raw wideband data
quality and
        accounting information on a contact period basis.

PARAMETERS:

RETURN VALUE:
Type = Void

Value          Description
-------------- ------------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Processing_Version_Info     I  Contains raw data processing version information
RDC_Acct                    I  Contains contact period time information

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_LPSQA
*****************************************************************
```

UNIT NAME: mac_ui_LPSThres

PURPOSE: This form allows users to view or update thresholds for RDPS, MFPS, or PCDS.

PARAMETERS:

RETURN VALUE:
Type = Void

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
Valid_MFP_Thres	I/O	MFP processing threshold
Valid_PCD_Thres	I/O	PCD processing threshold
Valid_RDP_Thres	I/O	RDP processing threshold

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
It is assumed that the operator will not change the LPS thresholds during processing

NOTES:

End Unit mac_ui_LPSThres
****

UNIT NAME:mac_ui_MainMenu

PURPOSE: This form provides capabilities for users to operate the LPS system via the menu items.

PARAMETERS:

RETURN VALUE:
Type = Void

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

Form Name: mac_ui_MainMenu

Purpose: This form uses a custom menu that is specifically built for the LPS system. It provides capabilities for the users to operate the LPS system via the menu items. The main menu includes SHUTDOWN, SETUP, TEST, CONTROL, MONITOR, FILES and REPORTS.

Enter Form Logic: Query and display the string id.

Exit Form Logic: Shutdown the LPS system

Special Form Logic:

Shutdowm menu option from the main menu

Event: Select "Shutdown" menu item
Action: Invoke mac_MainShutdownLPS to shutdown the LPS system
If LPS_SUCCESS is returned THEN Exit the form

Setup menu option from the main menu

Event: Select "LPS String Configuration..." menu item from the SETUP pull-down menu
Action: Invoke mac_ui_LPSConfig form

Event: Select "Data Processing Thresholds..." menu item from the SETUP pull-down menu
Action: Invoke mac_ui_LPSThres form

Event: Select "Data Processing Parameters..." menu item from the SETUP pull-down menu
Action: Invoke mac_ui_LPSParms form

Event: Select "Contact Schedules..." menu item from the SETUP pull-down menu
Action: Invoke mac_ui_ContSchedule form

Event: Select "Ingest IAS Parms..." menu item from the SETUP pull-down menu
Action: Invoke mac_ui_IngestIASParms form

```
# Control menu option from the main menu

 Event:  Select "Start Capture.." menu item from the CONTROL pull-down menu
 Action: Invoke mac_ui_StartCapture form

 Event:  Select "Stop Capture..." menu item from the CONTROL pull-down menu
 Action:  Invoke mac_ui_StopCapture form

 Event:  Select "Start Copy to Tape..." menu item from the CONTROL pull-down menu
 Action:  Invoke mac_ui_StartBackup form

 Event:  Select "Stop Copy to Tape..." menu item from the CONTROL pull-down menu
 Action:  Invoke mac_ui_StopBackup form

 Event:  Select "Start Copy from Tape..." menu item from the CONTROL pull-down menu
 Action:  Invoke mac_ui_StartRestage form

 Event:  Select "Stop Copy from Tape..." menu item from the CONTROL pull-down menu
 Action:  Invoke mac_ui_StopRestage form

 Event:  Select "Start Processing..." menu item from the CONTROL pull-down menu
 Action:  Invoke mac_ui_StartL0R form

 Event:  Select "Stop Processing..." menu item from the CONTROL pull-down menu
 Action:  Invoke mac_ui_StopL0R form

 Event:  Select "Start/Stop DAN Transfer..." menu item from the CONTROL pull-down menu
 Action:  Invoke mac_ui_DANTransState form

 Event:  Select "Start/Stop AutoCapture..." menu item from the CONTROL pull-down menu
 Action:  Invoke mac_ui_StartStopAutoCapture form

  # Files menu option from the main menu

 Event:  Select "Resend DAN..." menu item from the FILES pull-down menu
 Action:  Invoke mac_ui_ResendDAN form

 Event:  Select "Retain Files..." menu item from the FILES pull-down menu
 Action:  Invoke mac_ui_RetainFileSet form

 Event:  Select "Delete Files..." menu item from the FILES pull-down menu
 Action:  Invoke mac_ui_DeleteFileSet form

 Event:  Select "Start/Stop DDN Server..." menu item from the FILES pull-down menu
 Action:  Invoke mac_ui_StartStopDDNServer form

 # Monitor menu option from the main menu

 Event:  Select "Display LPS Journal File..." menu item from the MONITOR pull-down menu
 Action:  Invoke mac_ui_ExamLPSJournal form

 Event:  Select "Display Operation Messages..." menu item from the MONITOR pull-down menu
 Action:  Invoke mac_ui_OpsMsg form
```

Report menu item from the main menu

Event: Select "Data Receive Summary..." menu item from the REPORT pull-down menu
Action: Invoke mac_ui_DataRcvSum form

Event: Select "Level 0R QA..." menu item from the REPORT pull-down menu
Action: Invoke mac_ui_LPSQA form

Event: Select "File Transfer Summary..." menu item from the REPORT pull-down menu
Action: Invoke mac_ui_FileTransSum form

Test menu item from the main menu

Event: Select "Send Data..." menu item from the Test pull-down menu
Action: Invoke mac_ui_SendData form

Enhancements:

Limitations:

****

```
UNIT NAME mac_ui_MainShutdown

PURPOSE: Shutdown the LPS.

PARAMETERS:

RETURN VALUE:
Void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                    Use  Description
----------------------- ---  ---------------------------------------------------
--------
Process_Id              I    Database table containing Process Ids

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_MainShutdown
****************************************************************
```

```
UNIT NAME:mac_ui_OpsMsg

PURPOSE:  This form allows users to select the filter severity of the operation
                messages to be displayed in the operation's window.

PARAMETERS:

RETURN VALUE:
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
LPS_Configuration          I  Get string id from LPS Configuration table

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_OpsMsg
*****************************************************************
```

```
UNIT NAME:mac_ui_ResendDAN

PURPOSE: This form allows users to resend DANs to LP DAAC.

PARAMETERS:

RETURN VALUE:
Type = Void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LDT_File_Set_Info           I  LDT file set info

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_ResendDAN
****************************************************************
```

```
UNIT NAME:mac_ui_RetainFileSet

PURPOSE:  This form allows the operator to retain LPS output files of a contact
                period whose File_Set_Status in the LDT_File_Set_Info table is
not                   LDT_Done

PARAMETERS:

RETURN VALUE:  None
Type = Void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------------
LDT_File_Set_Info           I  Database table containing file set status

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_RetainFileSet
****************************************************************
```

```
UNIT NAME mac_ui_SendData

PURPOSE: This form allows the operator to select an on-line raw data file name
         from the list and to enter isolate flag info, and suspend L0R processing
         flag info. Then it invokes rdc_TestData to write data from the selected
         data file to a device.

PARAMETERS:

RETURN VALUE:
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                    Use  Description
----------------------- ---  ----------------------------------------------------
--------
RDC_Acct                I    Database table contining information about captured
raw data
Process_Id              I    Database table containing L0R prosesses' parent
process id

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_SendData
*****************************************************************
```

```
UNIT NAME: mac_ui_StartBackup

PURPOSE:  This form allows users to start backup an online raw data file to tape

PARAMETERS:

RETURN VALUE:
Type = Void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
RDC_Acct                    I  Raw Data Capture accounting table

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_StartBackup
*************************************************************
```

```
UNIT NAME:mac_ui_StartCapture

PURPOSE: This form allows users to start a raw data capture process.

PARAMETERS:

RETURN VALUE:
Type = Void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_StartCapture
*************************************************************
```

```
UNIT NAME mac_ui_StartL0R

PURPOSE: This form provides a list of on-line contact period data available for
L0R      processing. It allows users to select a data file and update the version
               number. Then it invokes mac_MainStartL0R to start the level zero
                     processing. The user can also choose if the selected raw
data file should           be deleted after being processed.

PARAMETERS:

RETURN VALUE:
Type = Void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
RDC_Acct                   I  List of avaiable contact period data to process

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_StartL0R
*****************************************************************
```

```
UNIT NAME:mac_ui_StartRestage

PURPOSE:  This form allows users to restage raw data from the removebla media to
disk

PARAMETERS:

RETURN VALUE:
Type = Void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
LPS_Configuration           I  LPS Configuration table

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_StartRestage
*****************************************************************
```

```
UNIT NAME mac_ui_StartStopAutoCapture

PURPOSE: This form allows the operator to start or stop the raw data automatic
         capture process.

PARAMETERS:

RETURN VALUE:
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                    Use  Description
----------------------- ---  ---------------------------------------------------
--------
Process_Id              I    Database table containing LPS process Id

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_StartStopAutoCapture
*****************************************************************
```

```
UNIT NAME mac_ui_StartStopDDNServer

PURPOSE: This form allows the operator to start or stop the DDN server process.

PARAMETERS:

RETURN VALUE:
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                    Use  Description
----------------------- ---  ---------------------------------------------------
Process_Id              I    Database table containing DDN server process id

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_StartStopDDNServer
****************************************************************
```

```
UNIT NAME: mac_ui_StopBackup

PURPOSE: To cancel a start tape copy request.

PARAMETERS:

RETURN VALUE:
Type = Void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_StopBackup
*************************************************************
```

```
UNIT NAME:  mac_ui_StopCapture

PURPOSE:  This form allows users to stop the raw data capture process.

PARAMETERS:

RETURN VALUE:
Type = Void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mac_ui_StopCapture
*************************************************************
```

UNIT NAME: mac_ui_StopL0RProc

PURPOSE: This form allows user to stop level Zero processing.

PARAMETERS:

RETURN VALUE: None
Type = Void

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
Process_Id	I	LPS process that is currently running on the system
RDC_Acct	I	RDC Accounting table

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

Enter Form Logic: Query and display the contact_sequence_id, file_version_num and parent_process_id from the process_id table and corresponding lps_hardware_string_id and contact start time of L0R process from the RDC_Acct table.

Exit Form Logic: Prohibit commit inquiry for all users.

Special Form Logic:

```
Event:  No record found
Action: Display a message indicating there are no L0R processes running
        and exit the form.

Event:  Select an entry and Press "OK" button
Action: Fork mac_MainStopL0R WITH
           mac_mainContSeqIdArg  = contact sequence ID,
           mac_mainfileVerNumArg = file version number, and
           mac_mainPidArg = process ID
        to stop the L0R process.

Event:  Press "CANCEL" button
Action: Exit form
```

Enhancements:

Limitations:

****

```
UNIT NAME:mac_ui_StopRestage

PURPOSE: This form allows user to stop a raw wideband data restage request

PARAMETERS:

RETURN VALUE:
Type = Void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

End Unit mac_ui_StopRestage
****************************************************************
```

```
UNIT NAME: mfp_AlignBands

PURPOSE: Align the band data.

PARAMETERS:
mfp_typeArg : data_in
mfp_bdLoadLocArg : data_out
mfp_scanDirArg : data_in

RETURN VALUE:
Type = void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
mfp_Parms       mfp_formatList      Structure       I   Array of band/det
                                                        combos in minor frames

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE
NOTES:

MFP_BAND6 = of the type band 6
MFP_MAXDETB6 = maximum number of detectors for band 6 data
MFP_MAXFMTDATA = maximum number of bytes for format 1 or 2

End Unit mfp_AlignBands
****************************************************************
```

UNIT NAME: mfp_CalL0rExtract

PURPOSE: To extract, deinterleave, and align the calibration data.
PARAMETERS:
mfp_calMscdArg : data_in
mfp_calFpArg : data_in
RETURN_CODE : control_out
mfp_calPcdStatSetFlgArg : control_in
mfp_calMjfTimeArg : data_in
mfp_calDetSubInfoArg : data_in
mfp_calNewSubIntvArg : data_in
mfp_calStartArg : data_inout
mfp_calScanDirArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Calibration data was successfully written to the file
LPS_FAILURE	Calibration data was unsuccessfully written to the file

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
CAL_File	O	The calibration file.

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_statusInfo	struct	I	status information
mfp_Global	mfp_calMemPtr	pointer	I	pointer to memory area that stores the cal data
mfp_Parms	mfp_sensorAlignInfo	integer	I	sensor alignment info
mfp_MjfQa	mfp_caduSyncErrCnt	integer	I	number of CADU sync Err
mfp_MjfQa	mfp_eolErrFlag	boolean	I	end of line error
mfp_MjfQa	mfp_partFillCnt	integer	I	number of frames filled
mfp_MjfQa	mfp_timeCodeErrCnt	integer	I	time code error

End Unit mfp_CalL0rExtract
****

UNIT NAME: mfp_CalWriteFile

PURPOSE: Writes Cal Data and File Header information to the calibration file.

PARAMETERS:
mfp_calDeintAlignDataArg : data_in
mfp_calNewSubIntvArg : data_in
mfp_calMjfTimeArg : data_in
mfp_calPcdStatSetFlgArg : data_in
RETURN_CODE : control_out
mfp_calFpArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Successful data write to file
LPS_FAILURE	Unsuccessful data write to file

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
CAL_File	O	Stores the calibration data

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_statusInfo	struct	I	Status Information
mfp_Global	mfp_calMemPtr	pointer	I	pointer to memory that stores the cal data
mfp_Global	mfp_calFileName	char	I	calibration file name
mfp_Global	mfp_spacecraftId			

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
NONE

NOTES: NONE

End Unit "mfp_CalWriteFile"
****

```
UNIT NAME:  mfp_ChckSplitMnf

PURPOSE: Determines if a split minor frame.

PARAMETERS:
mfp_chckLengthArg : data_in
mfp_chckAdjFactorArg : data_in
mfp_chckCaduLocArg : data_inout
mfp_chckSplitArg : control_out

RETURN VALUE:
Type = lpsBooleanType

Value          Description
-------------- -------------------------------------------------------------
LPS_TRUE       A return value indicating that it is a split minor frame
LPS_FALSE      A return value indicating that it is NOT a split minor frame

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
NONE

NOTES:
      NONE

End Unit mfp_ChckSplitMnf
****************************************************************
```

```
* UNIT NAME:   mfp_compareEol
*
* PURPOSE:   Locate a minor frame in the CADU buffer, compare it
*              to the 80-byte EOL pattern.
**
* UNIT TYPE:   (int) function
*

PARAMETERS:
mfp_VfyMfcArg : control_in
mfp_VfySyncOffsetArg : control_in
mfp_VfySyncOffsetArg : control_in
mfp_VfySyncIndexArg : control_in
RETURN_CODE : control_out

* RETURN VALUES:
*     5 = mf starts past end of mfp_avInfo[]
*     4 = partial match (trailing fill)
*     3 = partial match (leading fill)
*     2 = fill
*     1 = no match
*     0 = mf matches template
*    Note: These values become array indices in routine
*         mfp_FindMjfEol().  They are NOT to be changed.
*    Note: Use of names for these values reduces readability in
*         this routine, therefore the explicit values have been used.
*
* ARGUMENTS:
*    NAME                TYPE   use DESCRIPTION
*    ----------------------------------------------------------------------
*    sync_index           int    I  index for .mfp_avInfo[] where
*                                     the sync mf starts
*    sync_offset          int    I  offset in the CADU for start of sync
*    mfc                  int    I  minor frame count to check
*
* EXTERNAL VARIABLES:
*    SOURCE    NAME         TYPE  use  DESCRIPTION
*    ----------------------------------------------------------------------
*    "mfp_Parms.h"
*       mfp_MajSyncTol       int   I   limit for # bit errors allowed
*       mfp_MajSyncSize      int   I   min.# bytes needed in sync
*    "mfp_Global.h"
*       mfp_mjfVcduSet      struct
*         .mfp_vcduCnt       int   I   # of CADUs in array
*         .mfp_avInfo       struct     array of CADU info
*           .mfp_fillVcdu    bool  I   MFP-fill flag: T/F
*           .mfp_dataPtr    struct*    ptr to annotated CADU in shared mem.
*      [      .BCH_Corrected_Data uchar I   mission,status data bytes     ]
*
```

```
* EXTERNAL FILE/RECORD REFERENCES:
*    NAME                USE      DESCRIPTION
*    ----------------------------------------------------------------------
*    <stdio.h>        compile    Define I/O calls, structures
*    <stdlib.h>       compile    define system routine calls
*    "lps_?????.h"    compile    define LPS structures, #define's, etc.
*    "mfp_?????.h"    compile    define MFP structures, #define's, etc.
*    "mfp_Parms.h"    compile    define variables for tolerance checking
*    "mfp_Global.h"   compile    define list of CADUs in major frame
*
* EXTERNAL FUNCTIONS:   none
*
* Non_Portable Code:
*    1. Description:  machine/compiler-dependent automatic padding
*          at end of annotated CADU structure as defined in header
*          file " ".  Each annotated CADU must start on an 8-byte
*          boundary.  Since the annotated CADU as defined is not a
*          multiple of 8 bytes, the SGI compiler (most likely)
*          automatically inserts 3 bytes of pad before beginning the
*          next annotated CADU.  If LPS is transported to a single
*          4-byte or 8-byte word machine, the software should still
*          work.  If transported to a single byte-oriented machine,
*          the pad will disappear, producing smaller records which
*          may affect trouble files or other file access, but the
*          general processing should still work.  If LPS is
*          distributed to a mix of byte and 4/8-byte machines, data
*          passed from one machine to the next will be subject to 2
*          different physical interpretations of the same code, and
*          will fail.
*       Reason for use:  management re-write of the data structure
*          definition excluding explicit pad
*       Suggested porting procedure:  re-insert the explicit pad
*
* Written:
*    09/25/95   Gregory J Toth   NASA/GSFC/514.1
*            similar to mfp_FindMjfSync.c, but functionality for EOL
*
* Modified:
*    date      author    group       release/CCR
*        descriptions
*    -------   ---------------------------------------
*
* References:
*    1. Landsat-7 System - Data Format Control Book (DFCB),
*         Vol.4 - Wideband Data.
*         15 May 1995    Revision B
*         Martin Marietta Astro Space
*         [contains errors - see ref.2]
*    2. Hughes inter-departmental memo, Gary Trup to M.Giroux
*         subject: BEAST ASIC Minor Frame Format & Pointer Counter
*                  Definitions
*         Apr.19,1995
*         [describes data as implemented and verified in electronics]
*
```

```
* Assumptions:
*    1. The DFCB, with corrections per the Gary Trup memo, defines
*         the CADU and major frame data.  The Jackson & Tull memo of
*         June 7,1995 is incorrect and not usable.
*    2. The implementer is expected to be familiar with the relevant
*         satellite data structures.
*    3. The list of CADUs for the major frame must start with a
*         scan-direction flag change between the 1st 2 CADUs.  Either
*         or both of these may be MFP-generated fill, and will be
*         flagged as such by the generating routine.
*    4. The list of CADUs for the major frame must end with a data
*         (non-fill) CADU.  The following sub-conditions apply:
*         normal- the last CADU and the previous non-fill CADU
*                 contain a scan-direction change.  If any data
*                 CADUs are missing between them, fill CADUs will
*                 have been inserted by the calling routine.
*         other(EOC, VCID change, discontinuity) -
*                The end of the major frame will NOT be filled.
*                The last CADU is part of the current major frame.
*    5. VCIDs (except for the last CADU) are expected to be the same.
*    6. The mf_count and byte_offset in the CADU must be correct.
*         While this routine does not directly reference the VCDUcnt
*         (which does not have any correction scheme to protect it),
*         the fill generation is derived from it, and thus the
*         indexing for the CADU list.  The indexing is a critical
*         factor in the calculations of this routine.
*    7. Data pools will have the header file named according to the
*         data name specified in the Data Dictionary.  The pool will
*         directly define the variables listed in the Data Dictionary
*         structure.  The pool will NOT consist of a single variable
*         of the structure type in the Data Dictionary, as this will
*         cause inordinate extra complexity, coding difficulty, and
*         a significant decrease in maintainability.
*    8. Boolean variables are expected to use C standard definitions
*        for True/False.  Ad-hoc definitions that do not match the C
*        (and computing) standard will fail without warning.
*
*
***************************************************************************
```

UNIT NAME: mfp_CondenseDataGrp

PURPOSE: Condense a 40-repetition bit to a single bit

PARAMETERS:
mfp_dataGroupArg : data_in
mfp_bitValueArg : data_out

RETURN VALUE:
Type = (int) function

Value	Description
1	bit value =1
0	bit value =0
<0	negative of count of 1-bits, not consistent enough to declare a single 0/1 value

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
<stdio.h>		Define I/O calls, structures
<stdlib.h>		Define system routine calls

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_xxxx.h	mfp_MajVoteTol	int	I	min.# required bits to be the same

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mfp_CondenseDataGrp
****

UNIT NAME: mfp_db_GetParms

PURPOSE: Query the database and store the variables for MFPS parameters.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Database access was successful.
LPS_FAILURE	Database access failed.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
Valid_MFP_Parms	I	Valid MFP parameters.

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Parms	mfp_dataPeriod	lpsTimeStruct	O	Time between Mjfs
	mfp_EolTol	Natural	O	Max # of mnfs away from nominal to find EOL
	mfp_fillValue	char	O	1-byte fill character
	mfp_MajSyncSize	Natural	O	Min # of bytes for sync acceptance
	mfp_MajSyncTol	Natural	O	Min # of bit errors
	mfp_majVoteTol	Natural	O	Min # of bits
	mfp_maxAlignValue	Natural	O	Max # for align values
	mfp_maxCaduCnt	Natural	O	Max # Cadus/MJF
	mfp_partMnfTol	Natural	O	Min # of minor frames
	mfp_sensorAlignInfo	Struct	O	Alignment values
	mfp_subIntvDelta	Real	O	Min time gap to declare a subinterval
	mfp_timeRangeTol	lpsTimeStruct	O	Max length of time that could possibly be legitimate

End Unit mfp_db_GetParms
****

UNIT NAME: mfp_db_GetThres

PURPOSE: Query the database and store the variables for MFPS thresholds.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = void

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
Valid_MFP_Thres	I	Valid MFP threshold values.

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Thres	mfp_caduSeqErrThr	Natural	O	Cadu seq error limit
	mfp_syncThr	Natural	O	Max # Sync errors
	mfp_mnfCtrThr	Natural	O	Max # Mnf Ctr error limit
	mfp_eolThr	Natural	O	EOL error limit
	mfp_tcThr	Natural	O	Time Code error limit
	mfp_fullMjfThr	Natural	O	Max # of fully filled major frames (#missing)
	mfp_partMjfThr	Natural	O	Max # of partially filled major frames
mfp_ThresDefault	mfp_caduSeqErrThr	Natural	I	Cadu seq error limit
	mfp_syncThr	Natural	I	Max # Sync errors
	mfp_mnfCtrThr	Natural	I	Max # Mnf Ctr error limit
	mfp_eolThr	Natural	I	EOL error limit
	mfp_tcThr	Natural	I	Time Code error limit
	mfp_fullMjfThr	Natural	I	Max # of fully filled major frames (#missing)
	mfp_partMjfThr	Natural	I	Max # of partially filled major frames

End Unit mfp_db_GetThres
****

```
UNIT NAME: mfp_db_InsertSubIntv

PURPOSE: Inserts the subinterval information into the database.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Data was inserted into the database
LPS_FAILURE    Data was unable to be inserted into the database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Sub_Intv                   I/O Subinterval information.

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
mfp_Global      mfp_subIntv          Structure        I  Subinterval info

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:

End Unit mfp_db_InsertSubIntv
****************************************************************
```

```
UNIT NAME: mfp_db_InsertSubQa

PURPOSE: Inserts subinterval QA information into the database.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    The routine completed successfully
LPS_FAILURE    The routine did not complete successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
MFP_Acct                   O     MFPS subinterval quality and accounting DB

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
mfp_SubQa       mfp_subQa            mfpSubQaStruct  I   Subinterval Q&A
mfp_Global      mfp_subIntvClose     struct          I   Sub Intv info

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mfp_db_InsertSubQa
*****************************************************************
```

```
UNIT NAME: mfp_db_UpdateSubIntv

PURPOSE: Updates the subinterval stop time.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Updating the database was successful.
LPS_FAILURE    Failed to update or access the database.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Sub_Intv                  O   Update the stop time.

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
mfp_Global      mfp_subIntv         Structure       O   Local subintv info

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:

End Unit mfp_db_UpdateSubIntv
*****************************************************************
```

UNIT NAME: mfp_Deint

PURPOSE: Deinterleave part of one minor frame.

PARAMETERS:
mfp_arrayStarterArg : data_in
mfp_arrayMaxArg : data_in
mfp_splitArg : control_in
mfp_scanDirArg : data_in
mfp_caduLocArg : data_inout
mfp_bdLoadLocArg : data_inout

RETURN VALUE:
Type = void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES: NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:
This routine can be used for all bands and pan data. mfp_arrayStarter is necessary for band 6 data. The detectors alternate every other minor frame. The detector will be either 1 or 5.

End Unit mfp_Deint
****

```
  UNIT NAME:   mfp_ExtractMnf

  PURPOSE:   Extract 1-22 minor frames from 1 or 2 CADUs in the
             major frame VCDUset, if/as present

PARAMETERS:
RETURN_CODE : control_out
mfp_VfyStartMfcArg : control_in
mfp_VfyNumMfArg : control_in
mfp_VfySyncOffsetArg : control_in
mfp_VfySyncIndexArg : control_in
mfp_VfyMfBytesArg : control_out
mfp_VfyMfBufArg : data_out

  RETURN VALUE:
  Type = RETURN_CODE

  Value            Description
  ---------------  -----------------------------------------------------
  MFP_MISSING      missing,
  MFP_COMPLETE     complete,
  MFP_PARTIAL_TF   partial(start of data, trailing fill),
  MFP_PARTIAL_LF   partial(end of data, leading fill)

  EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
  None

  EXTERNAL VARIABLES:
  SOURCE       NAME           TYPE                use DESCRIPTION
  ----------   -------------- ------------------- --- ----------------
  mfp_Global   mfp_mjfVcduSet mfpMjfVcduSetStruct  I  Major frame VCDU set

 ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

     1. The minor frame(s) requested are expected to refer to the
         current major frame.  The calling routine is expected to
         ensure that the requested frame(frames) does(do) not extend
         into the sync of the next major frame or past a
         discontinuity between CADUs at the end of the VCDUset.
     2. Boolean variables are expected to use C standard definitions
         for True/False.  Ad-hoc definitions that do not match the C
         (and computing) standard will fail without warning.

 NOTES:
    LANGUAGE:  ANSI C
    UNIT TYPE:   (int) function

End Unit mfp_ExtractMnf
*****************************************************************
```

UNIT NAME: mfp_FillBand6

PURPOSE: Fill the Aligned band area for band 6 only.

PARAMETERS:
mfp_bdLoadLocArg : data_inout
mfp_scanDirArg : data_in

RETURN VALUE:
Type = void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Parms	mfp_fillValue	char	I	Fill value for aligning
mfp_Parms	mfp_sensorAlignInfo	Structure	I	Alignment Information
mfp_Parms	mfp_formatList	Structure	I	Array of band/det combos in minor frames

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE
NOTES:
Note the differences between forward and reverse scan. For the forward scan, this routine will return the location in lps_AlignedBands after the last filled byte from the west end, but for the reverse scan, data will be placed starting from the east end.

MFP_MAXDETB6 = the maximum number of detectors for band 6 data
MFP_FORWARD (MFP_REVERSE) = the values of the scan direction
MFP_EAST (MFP_WEST) = the geographical direction irrespective of scan direction
MFP_FULL_BANDLENGTH = the size in bytes for one set of band data, including the aligned area.

End Unit mfp_FillBand6
****

UNIT NAME: mfp_FillMostBands

PURPOSE: Fill the Aligned band area for all bands except band 6.

PARAMETERS:
mfp_scanDirArg : data_in
mfp_bdLoadLocArg : data_inout

RETURN VALUE:
Type = void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Parms	mfp_fillValue	char	I	Fill value for aligning
mfp_Parms	mfp_sensorAlignInfo	Structure	I	Alignment Information
mfp_Parms	mfp_formatList	Structure	I	Array of band/det combos in minor frames

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE
NOTES:
Note the differences between forward and reverse scan. For the forward scan, this routine will return the location in lps_AlignedBands after the last filled byte from the west end, but for the reverse scan, data will be placed starting from the east end.

MFP_PARITYSIZE = the number of bytes in format 1 or 2 for odd or even data
MFP_FORWARD (MFP_REVERSE) = the values of the scan direction
MFP_EAST (MFP_WEST) = the geographical direction irrespective of scan direction
MFP_FULL_BANDLENGTH = the size in bytes for one set of band data, including the aligned area.

End Unit mfp_FillMostBands
****

```
* UNIT NAME:   mfp_FindMjfEol.c
*
* PURPOSE:   Find the EOL minor frame pair for the major frame
*
* LANGUAGE:  ANSI C
*
* UNIT TYPE:   (int) function
*

PARAMETERS:
mfp_VfySyncOffsetArg : control_in
mfp_VfySyncIndexArg : control_in
mfp_VfyEolmfcArg : control_out
mfp_VfyEolOffsetArg : control_out
mfp_VfyEolIndexArg : control_out
RETURN_CODE : control_out

* RETURN VALUES:
*    +1 =  good EOL found
*     0 =  EOL not found
*    -1 =  telemetry/programming errors
*
* ARGUMENTS:
*    NAME                TYPE   use DESCRIPTION
*    ---------------------------------------------------------------------
*    sync_index           int    I  index for .mfp_avInfo[] where
*                                     the sync mf starts
*    sync_offset          int    I  offset in the CADU for start of sync
*    eol_index            int    O  index for .mfp_avInfo[] where
*                                     the 1st EOL mf starts
*    eol_offset           int    O  offset in the CADU for start of EOL
*    eol_mfc              int    O  minor frame count for 1st EOL mf
*
*
* EXTERNAL VARIABLES:
*    SOURCE    NAME         TYPE  use  DESCRIPTION
*    ---------------------------------------------------------------------
*    "mfp_Parms.h"
*      mfp_EolTol            int   I   +/- value for setting limits
*                                        around nominal EOL start
*                                        (mf#6320) to cut off EOL
*                                        checking.
*
* EXTERNAL FILE/RECORD REFERENCES:
*    NAME               USE      DESCRIPTION
*    ---------------------------------------------------------------------
*    <stdio.h>        compile    Define I/O calls, structures
*    <stdlib.h>       compile    define system routine calls
*    "lps_?????.h"    compile    define LPS structures, #define's, etc.
*    "mfp_?????.h"    compile    define MFP structures, #define's, etc.
*    "mfp_Parms.h"    compile    define variables for tolerance checking
*
* EXTERNAL FUNCTIONS:   none
*
```

```
* Non_Portable Code:
*    1. Description:  machine/compiler-dependent automatic padding
*          at end of annotated CADU structure as defined in header
*          file " ".  Each annotated CADU must start on an 8-byte
*          boundary.  Since the annotated CADU as defined is not a
*          multiple of 8 bytes, the SGI compiler (most likely)
*          automatically inserts 3 bytes of pad before beginning the
*          next annotated CADU.  If LPS is transported to a single
*          4-byte or 8-byte word machine, the software should still
*          work.  If transported to a single byte-oriented machine,
*          the pad will disappear, producing smaller records which
*          may affect trouble files or other file access, but the
*          general processing should still work.  If LPS is
*          distributed to a mix of byte and 4/8-byte machines, data
*          passed from one machine to the next will be subject to 2
*          different physical interpretations of the same code, and
*          will fail.
*       Reason for use:  management re-write of the data structure
*          definition excluding explicit pad
*       Suggested porting procedure:  re-insert the explicit pad
*
* Written:
*    06/07/95   Gregory J Toth   NASA/GSFC/563.2
*
* Modified:
*    date      author    group       release/CCR
*        descriptions
*    -------   ---------------------------------------
*    9/19/95   Greg Toth   NASA/GSFC/514.1
*        re-designed for detailed design
*           a) respond to re-design of MjfVcduSet
*           b) respond to corrected minor-frame-count specs.
*        initial coding (subject to structure implementation
*           in header files)
*    10/18/95  Greg Toth   NASA/GSFC/514.1
*        prolog touch-up
*
* References:
*    1. Landsat-7 System - Data Format Control Book (DFCB),
*         Vol.4 - Wideband Data.
*         15 May 1995    Revision B
*         Martin Marietta Astro Space
*         [contains errors - see ref.2]
*    2. Hughes inter-departmental memo, Gary Trup to M.Giroux
*         subject: BEAST ASIC Minor Frame Format & Pointer Counter
*                  Definitions
*         Apr.19,1995
*         [describes data as implemented and verified in electronics]
*
*
```

```
Assumptions:
*    1. The DFCB, with corrections per the Gary Trup memo, defines
*         the CADU and major frame data.  The Jackson & Tull memo of
*         June 7,1995 is incorrect and not usable.
*    2. The implementer is expected to be familiar with the relevant
*         satellite data structures.
*    3. The list of CADUs for the major frame must start with a
*         scan-direction flag change between the 1st 2 CADUs.  Either
*         or both of these may be MFP-generated fill, and will be
*         flagged as such by the generating routine.
*    4. The list of CADUs for the major frame must end with a data
*         (non-fill) CADU.  The following sub-conditions apply:
*         normal- the last CADU and the previous non-fill CADU
*                 contain a scan-direction change.  If any data
*                 CADUs are missing between them, fill CADUs will
*                 have been inserted by the calling routine.
*         other(EOC, VCID change, discontinuity) -
*                The end of the major frame will NOT be filled.
*                The last CADU is part of the current major frame.
*    5. VCIDs must be the same.
*    6. The mf_count and byte_offset in the CADU must be correct.
*         While this routine does not directly reference the VCDUcnt
*         (which does not have any correction scheme to protect it),
*         the fill generation is derived from it, and thus the
*         indexing for the CADU list.  The indexing is a critical
*         factor in the calculations of this routine.
*    7. Data pools will have the header file named according to the
*         data name specified in the Data Dictionary.  The pool will
*         directly define the variables listed in the Data Dictionary
*         structure.  The pool will NOT consist of a single variable
*         of the structure type in the Data Dictionary, as this will
*         cause inordinate extra complexity, coding difficulty, and
*         a significant decrease in maintainability.
*     8. It is expected that (6320 - mfp_eolTol) > 6 .  No checking
*         is performed.
*

 End Unit  mfp_FindMjfEol
*****************************************************************
```

```
* UNIT NAME:   mfp_FindMjfSync
*
* PURPOSE:   Find the start/end sync minor frame of the major frame
*               in the VCDU buffer
*
* LANGUAGE:  ANSI C
*
* UNIT TYPE:   (int) function
*

PARAMETERS:
mfp_VfySyncIndexArg : control_out
mfp_VfySyncBegEndArg : control_in
mfp_VfySyncOffsetArg : control_out
RETURN_CODE : control_out

* RETURN VALUES:
*    +1 =  good sync found
*     0 =  sync fill or part fill
*    -1 =  invalid CADU:
*            a) bit_errs > limit:
*                 other data instead of sync
*                 mfc was bad in CADU
*            b) bad control value(s) indicated:
*                    BCH uncorrectable for offset
*                    BCH uncorrectable for mission-data AND bad 0-fill
*                    offset > 168 :
*                 data inversion during CADU
*                 bitslip during CADU
*    -2 =  sync starts before mfp_avInfo[0]
*            ( possible programming error in prior routine )
*
* ARGUMENTS:
*    NAME                TYPE   use DESCRIPTION
*    ---------------------------------------------------------------------
*    beg_end_flag         int    I  flag for checking beginning or end
*                                      of major frame for sync
*                                      0 = beginning,
*                                      1 = end
*    sync_index           int    O  index for .mfp_avInfo[] where
*                                     the sync mf starts
*    sync_offset          int    O  offset in the CADU for start of sync
*
*
```

```
* EXTERNAL VARIABLES:
*    SOURCE    NAME         TYPE  use  DESCRIPTION
*    ----------------------------------------------------------------------
*    "mfp_Parms.h"
*         mfp_MajSyncTol     int   I   limit for # bit errors allowed
*         mfp_MajSyncSize    int   I   min.# bytes needed in sync
*    "mfp_Global.h"
*         mfp_mjfVcduSet    struct
*           .mfp_vcduCnt     int   I   # of CADUs in array
*           .mfp_avInfo     struct     array of CADU info
*             .mfp_fillVcdu  bool  I   MFP-fill flag: T/F
*             .mfp_dataPtr  struct*    ptr to annotated CADU in shared mem.
*               .Mission_Data uchar I  mission,status data bytes
*               .Data_Pointer uchar I  data pointer (offset) to start of
*                                          1st full min.frm in CADU
*               .BCH_Mission_Quality_Indicator
*                             uchar I  quality flag for .Mission_Data
*               .BCH_Pointer_Quality_Indicator
*                             uchar I  quality flag for .Data_Pointer
*
* EXTERNAL FILE/RECORD REFERENCES:
*    NAME               USE      DESCRIPTION
*    ----------------------------------------------------------------------
*    <stdio.h>        compile    Define I/O calls, structures
*    <stdlib.h>       compile    define system routine calls
*    "lps_?????.h"    compile    define LPS structures, #define's, etc.
*    "mfp_?????.h"    compile    define MFP structures, #define's, etc.
*    "mfp_Parms.h"    compile    define variables for tolerance checking
*    "mfp_Global.h"   compile    define list of CADUs in major frame
*
* EXTERNAL FUNCTIONS:   none
*
* Non_Portable Code:
*    1. Description:  machine/compiler-dependent automatic padding
*          at end of annotated CADU structure as defined in header
*          file " ".  Each annotated CADU must start on an 8-byte
*          boundary.  Since the annotated CADU as defined is not a
*          multiple of 8 bytes, the SGI compiler (most likely)
*          automatically inserts 3 bytes of pad before beginning the
*          next annotated CADU.  If LPS is transported to a single
*          4-byte or 8-byte word machine, the software should still
*          work.  If transported to a single byte-oriented machine,
*          the pad will disappear, producing smaller records which
*          may affect trouble files or other file access, but the
*          general processing should still work.  If LPS is
*          distributed to a mix of byte and 4/8-byte machines, data
*          passed from one machine to the next will be subject to 2
*          different physical interpretations of the same code, and
*          will fail.
*       Reason for use:  management re-write of the data structure
*          definition excluding explicit pad
*       Suggested porting procedure:  re-insert the explicit pad
*
* Assumptions:
*    1. The DFCB, with corrections per the Gary Trup memo, defines
*         the CADU and major frame data.  The Jackson & Tull memo of
*         June 7,1995 is incorrect and not usable.
*    2. The implementer is expected to be familiar with the relevant
```

```
*         satellite data structures.
*    3. The list of CADUs for the major frame must start with a
*         scan-direction flag change between the 1st 2 CADUs.  Either
*         or both of these may be MFP-generated fill, and will be
*         flagged as such by the generating routine.
*    4. The list of CADUs for the major frame must end with a data
*         (non-fill) CADU.  The following sub-conditions apply:
*         normal- the last CADU and the previous non-fill CADU
*                 contain a scan-direction change.  If any data
*                 CADUs are missing between them, fill CADUs will
*                 have been inserted by the calling routine.
*         other(EOC, VCID change, discontinuity) -
*                The end of the major frame will NOT be filled.
*                The last CADU is part of the current major frame.
*    5. VCIDs must be the same.
*    6. The mf_count and byte_offset in the CADU must be correct.
*         While this routine does not directly reference the VCDUcnt
*         (which does not have any correction scheme to protect it),
*         the fill generation is derived from it, and thus the
*         indexing for the CADU list.  The indexing is a critical
*         factor in the calculations of this routine.
*    7. Data pools will have the header file named according to the
*         data name specified in the Data Dictionary.  The pool will
*         directly define the variables listed in the Data Dictionary
*         structure.  The pool will NOT consist of a single variable
*         of the structure type in the Data Dictionary, as this will
*         cause inordinate extra complexity, coding difficulty, and
*         a significant decrease in maintainability.
*    8. Boolean variables are expected to use C standard definitions
*        for True/False.  Ad-hoc definitions that do not match the C
*        (and computing) standard will fail without warning.
*
* Note:
*    The sync will start somewhere in the 1st CADU.  The scan bit
*    of the 1st CADU is set before the telemetry is inserted, so
*    even if the sync starts in byte 0, the scan bit will reflect
*    the last major frame even though the mfc will be 0.  Using the
*    2nd CADU will point to the correct byte for the sync start.
*    If the sync starts at the end of the 1st CADU and crosses into
*    the 2nd CADU, the mfc will be 0 in the 2nd CADU and the scan
*    bit will reflect the new major frame (different from the 1st
*    scan bit).  The sync can never start in the 2nd CADU.  Thus,
*    the mfc from the 2nd CADU never needs adjustment for the sync
*    interrupt.
*
End Unit  mfp_FindMjfSync
************************************************************
```

UNIT NAME: mfp_L0RFilesGen

PURPOSE: Controls the process of extracting the MSCD and calibration data into their respective files.

PARAMETERS:
mfp_l0rCalFileInfoArg : data_out
mfp_l0rMscdFileInfoArg : data_out
mfp_l0rNewSubIntvArg : data_in
mfp_l0rCalMscdArg : data_inout
mfp_l0rMjfTimeArg : data_in
mfp_l0rDetSubInfoArg : data_in
mfp_l0rPcdStatSetFlgArg : control_in
RETURN_CODE : control_out
mfp_l0rScanDirArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_FAILURE	Failure to successfully open,close, or create a file/filename
LPS_SUCCESS	Successfully opening,closing, or creating a file/filename

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
MSCD_File	O	Stores the MSCD data.
CAL_File	O	Stores the deinterleaved and aligned calibration data.

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_calFileName	char	O	The name of the calibration file.
mfp_Global	mfp_mscdFileName	char	O	The name of the MSCD file.
mfp_subIntv	lps_subIntvId	Sub_Intv_Sequence_Id	I	Sub Int Identifier

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. This unit will fail the first time it is called if the condition for a subinterval does not equal the new subinterval condition (the start of the first subinterval or a subinterval after a data break) or a normal subinterval (the end of one subinterval and the start of a new subinterval).

NOTES:
NONE

End Unit "mfp_L0RFilesGen"
****

UNIT NAME: mfp_Main

PURPOSE: Control Level 0R Processing for a major frame.

PARAMETERS:
mfp_mainContactIdArg : data_in
mfp_mainFileVerNumArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	MFPS did something successfully. (started up or finished)
LPS_FAILURE	MFPS flunked.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
mfp_FailedMjfs	O	Temporary file storage for VCDUs that fail major frame validation.

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_ShmAddList	mfpShmAddListStruct	IO	Structure holding Shm Block addresses

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

1. MFPS will retain up to 2 shared memory blocks for next iteration out of the blocks locked by it while returning the blocks to shared memory for use by RDPS.

NOTES:

End Unit mfp_Main
****

```
UNIT NAME: mfp_MainAdd2Set

PURPOSE: To append the current VCDU to the mfp_mjfVcduSet and to check whether
         the mfp_mjfVcduSet is complete.
PARAMETERS:
RETURN_CODE : control_out
mfp_mainDetSubInfoArg : data_out
mfp_mainAvInfoArg : data_inout
mfp_mainPreAvInfoArg : data_inout

RETURN VALUE: RETURN_CODE
Type =RETURN_CODE

Value              Description
--------------     ------------------------------------------------------------
MFP_NO_CHANGE      The VCDU added to Mjf VCDU set; None of above
#
LPS_FAILURE        The call did not complete successfully
MFP_EOC            End of Contact detected
MFP_MJFSET_FULL    The VCDU set has overflowed
MFP_VCID_CHANGE    The VCDU added to Mjf VCDU set but VCID change detected
MFP_NORMAL         The VCDU added to Mjf VCDU set and Scan bit change detected
MFP_DATA_BREAK     The VCDU added to Mjf VCDU set but lost track of Mjf Synch
MFP_END_FILLED     The VCDU added to Mjf VCDU set, Scan bit change detected but
                        Missing CADUs before last CADU in the set(Not filled)

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:
Source      Name            Type                  Use Description
----------  --------------  ------------------    --- ----------------------
mfp_global  mfp_mjfVcduSet  mfpMjfVcduSetStruct   IO  VCDUs in a Mjf set
mfp_Parms   mfp_parms           struct            I   MFPS parameters

lps_shm_RDPMFP lps_annCadu lpsAnnotatedCaduStruct I   Ann Cadu Data

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mfp_MainAdd2Set
*****************************************************************
```

UNIT NAME: mfp_MainBandGen

PURPOSE: Control generation of aligned bands.
PARAMETERS:
mfp_mainMjfPrevTimeArg : data_in
mfp_mainMjfTimeArg : data_in
mfp_mainBandsArg : data_in
mfp_mainScanDirArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Aligned bands completed successfully.
LPS_FAILURE	Aligned bands failed to complete successfully.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_statusInfo	char	I	2-byte status info.
mfp_Global	mfp_subIntv	struct	I	Using subinterval Id
mfp_MjfQa	mfp_fullFillCnt	int	I	# of fully filled Mjfs

NOTES: The total number of alignment bytes is 194. 6313 is used as the average number of mnfs/mjf. Therefore, each band/detector combo will be 6504 bytes long. Band 6 will be ~3912 bytes long (194 alignment bytes + 7436/2 mnfs).

MFP_BAND6 = of type band 6
MFP_FMT1 = of type format 1
MFP_FMT2 = of type format 2
MFP_MAXB6BYTES = number of bytes per minor frame for band 6 data
MFP_FMTDATA = number of bytes peer minor frame for format 1 or 2
MFP_AVGNUM_MNFS = average number of minor frames
MFP_ADJFACTOR_B6 = 80
MFP_ADJFACTOR_ALLBDS = 5

End Unit mfp_MainBandGen
****

UNIT NAME: mfp_MainCheckTcFrames

PURPOSE: Checks the time code minor frames for missing data words.

PARAMETERS:
mfp_mainMjfTcFramesArg : data_in
mfp_mainLastFullDataWordArg : data_out
mfp_mainLastGoodMnfArg : data_out
mfp_mainFirstFullDataWordArg : data_out
mfp_mainFirstGoodMnfArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	The actual data groups from the first and last time code minor frames are valid.
LPS_FAILURE	The actual data groups from the first and last time code minor frames are invalid.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES: NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: If the first and last minor frames fail under any conditions, the time code error count will be incremented. This error will not stop the processing of the major frame and major frame time.

NOTES: NONE

End Unit mfp_MainCheckTcFrames
****

UNIT NAME: mfp_MainChkMissing

PURPOSE: To determine missing VCDUs, if any, before the current VCDU and, if possible, correct the VCDU set counters.

PARAMETERS:
mfp_mainVcduSetArg : data_in
mfp_mainAvInfoArg : data_inout
mfp_mainPreAvInfoArg : data_inout
mfp_mainSeqFlagArg : control_out

RETURN VALUE:
Type = Void

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
lps_shm_RDPMFP	lps_annCadu	lpsAnnotatedCaduStruct	O	Ann Cadu Data

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

1. If the VCDU counter of mfp_mainAvInfoArg is greater than the VCDU counter of mfp_mainPreAvInfoArg (taking Rollover into account), the probability of the VCDU counters being correct is high.

2. If the BCH_Mission_Quality_Indicator of a CADU is NOT LPS_UNCORRECTABLE, the Mnf Counter for the CADU is assumed to be correct with acceptable risk

3. If the RS_Annotation of a CADU is LPS_NOERROR, the VCDU Counter for the CADU is assumed to be correct with acceptable risk.(Even though the VCDU counter is not protected by RS code, it is in the middle of data protected by the RS code and is therefore likely to be correct if the code protected by the RS code is correct)

4. Missing VCDUs don't represent VCDU counter error.

End Unit mfp_MainChkMissing
****

```
UNIT NAME: mfp_MainChkSeq

PURPOSE: To find,if possible, a valid CADU in the shared memory following the
         latest CADU in the tossed VCDU set  with continuity between the two
         CADUs; otherwise return with flag indicating failure to do so.
         (MFP_NOT_DONE)
PARAMETERS:
mfp_mainNumDisArg : data_inout
mfp_mainScanBitArg : control_in
mfp_mainAvInfoArg : data_inout
mfp_mainPreAvInfoArg : data_inout
mfp_mainVcduSetArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
MFP_NOT_DONE       The desired CADU not found;
MFP_DATA_BREAK     found discontinuity between CADUs.
#
LPS_FAILURE        The routine did not complete sucessfully
MFP_TOSSEDSET_FULL Desired CADU found, but exit since Tossed VCDU buffer is full
MFP_NO_CHANGE      Desired CADU found, but NO SCANBIT_FLIP, NO full tossed set

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
mfp_Global  mfp_tossedVcduSet   mfpMjfVcduSetStruct  IO VCDUs without a Mjf
mfp_Global  mfp_mjfVcduSet      mfpMjfVcduSetStruct  O  VCDUs in a Mjf
mfp_Global  mfp_nextAvInfo      mfpAvInfoStruct      IO Temp hold for a CADU
mfp_Global  mfp_nextAvInfoFlag  lpsBooleanType       IO  True if mfp_nextInfo
                                                         is filled;
lps_shm_RDPMFP lps_annCadu    lpsAnnotatedCaduStruct I  Ann Cadu Data

End Unit mfp_MainChkSeq
*****************************************************************
```

```
UNIT NAME:mfp_MainChkSeqMjf

PURPOSE: To find, if possible, a valid CADU in the shared memory following the
         latest CADU in the Mjf VCDU set with continuity between the two CADUs;
         Otherwise return with flag indicating failure to do so.
         (MFP_NOT_DONE)
PARAMETERS:
mfp_mainNumDisArg : data_inout
mfp_mainPreAvInfoArg : data_inout
RETURN_CODE : control_out
mfp_mainVcduSetArg : data_in
mfp_mainAvInfoArg : data_inout
mfp_mainScanBitArg : control_in

RETURN VALUE:
Type =RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
MFP_NOT_DONE       The desired CADU not found;
MFP_DATA_BREAK     Lost track of Mjf Synch
#
LPS_FAILURE        The call did not complete successfully
MFP_MJFSET_FULL    Desired CADU found, but exit since the Mjf VCDU set is full
MFP_END_FILLED     Desired CADU found, but there are Missing CADUs before last
                   CADU in the set(Not filled)
MFP_NO_CHANGE      Desired CADU found, but none of the above.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:
Source      Name                Type                  Use Description
----------- ------------------- ---------------       --- --------------------
mfp_Parms   mfp_parms           struct                I  MFPS parameters
mfp_Global  mfp_mjfVcduSet      mfpMjfVcduSetStruct   IO VCDUs in a Mjf
mfp_Global  mfp_nextAvInfo      mfpAvInfoStruct       IO Temp hold for a CADU
mfp_Global  mfp_nextAvInfoFlag  lpsBooleanType        O  True if mfp_nextInfo
                                                          is filled;
lps_shm_RDPMFP lps_annCadu     lpsAnnotatedCaduStruct I Ann Cadu Data

End Unit mfp_MainChkSeqMjf
*************************************************************
```

UNIT NAME: mfp_MainCleanup

PURPOSE: This routine deallocates shared memory areas and release key ids for the shared memory.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Cleanup complete successfully
LPS_FAILURE	Cleanup failed to shut down properly

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
CAL_File	I	Calibration file
MSCD_File	I	MSCD file

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_calFileName	String	I	Calibration file name
	mfp_mscdFileName	String	I	MSCD file name
	lps_mfps_pcdskey	lpsShmKeyType	I	MFPS/PCDS shm key
	lps_mfps_idpskey	lpsShmKeyType	I	MFPS/IDPS shm key
	lps_rdps_mfpskey	lpsShmKeyType	I	RDPS/MFPS shm key

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:

End Unit mfp_Cleanup
****

UNIT NAME: mfp_MainColVcduQA

PURPOSE: To update VCDU quality and accounting information for major frame and tossed VCDU sets

PARAMETERS:
mfp_mainQaSetArg : data_in
mfp_mainVcduSetArg : data_in
mfp_mainMjfSetEndArg : control_in

RETURN VALUE:
Type = void

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_MjfQa	mfp_mjfQa	mfpMjfQaStruct	IO	QA totals on Mjf basis
mfp_MjfQa	mfp_holdQa	mfpMjfQaStruct	IO	QA totals on hold
mfp_Global	mfp_mjfVcduSet	mfpMjfVcduSetStruct	I	VCDUs in a Mjf set
mfp_Global	mfp_tossedVcduSet	mfpMjfVcduSetStruct	I	VCDUs not belonging to a Mjf
lps_shm_RDPMFP	lps_annCadu	lpsAnnotatedCaduStruct	I	Ann Cadu Data

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. 'CADU polarity' and 'CADU polarity change' QA statistics are aggregated for non-flywhell CADUs only.

NOTES:
1. curPolarity is a local static variable holding the polarity of the last CADU used for QA aggregation.
2. firstTimeThru is a local static variable set to LPS_TRUE at startup.

End Unit mfp_MainColVcduQA
****

```
UNIT NAME:mfp_mainCompScanDir

PURPOSE: To compare the scan direction bits of the two arguments.

PARAMETERS:
mfp_mainPreAvInfoArg : data_inout
mfp_mainAvInfoArg : data_inout
RETURN_CODE : control_out

RETURN VALUE:RETURN_CODE
Type =lpsBooleanType

Value          Description
-------------- -------------------------------------------------------------
LPS_TRUE       Scan direction changed
LPS_FALSE      Scan direction did not change

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -----------------------------------------------

EXTERNAL VARIABLES:
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. If one or both scan direction bits being compared are not known, then it is
   assumed that the scan direction bit has not been changed.

NOTES:

End Unit "mfp_MainCompScanDir"
*************************************************************
```

UNIT NAME: mfp_MainDetermineSub

PURPOSE: Determine subinterval.

PARAMETERS:
mfp_mainDetSubInfoArg : data_in
mfp_mainMjfTimeArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_FAILURE	Database access has failed. Exit from MFPS.
MFP_NOSUB	No subinterval declared.
MFP_EOC	The end of the contact period.
MFP_NORMSUB	The end of one subinterval and the beginning of the next.
MFP_NEWSUB	The beginning of a subinterval only.
MFP_EOC_NOMJF	The end of contact, but no major frame is being processed.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_subIntv	Structure	I	Subinterval info
mfp_Global	mfp_subIntvClose	Structure	O	Subint info saved for closing out the Q&A with the previous stop time.

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
A new subinterval will not have an estimated time.
The subinterval ID will not be zero.

NOTES: NONE

End Unit mfp_MainDetermineSub
****

UNIT NAME: mfp_MainDetermineTimeRange

PURPOSE: Determine the subinterval gap.

PARAMETERS:
mfp_mainMjfTimeArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
MFP_NOSUB	No subinterval declared.
MFP_NORMSUB	The end of one subinterval and the beginning of the next.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_subIntv	structure	I	Use the stop time
mfp_Parms	mfp_subIntvDelta	lpsTimeStruct	I	Minimum time range for a subinterval gap
mfp_Parms	mfp_dataPeriod	lpsTimeStruct	I	Time between Mjfs
mfp_MjfQa	mfp_mjfQa	mfpMjfQaStruct		
	.mfp_fullFillCnt	int	I	Number of missing Mjfs before the current Mjf

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:

End Unit mfp_MainDeterminTimeRange
****

UNIT NAME: mfp_MainEstimateTime

PURPOSE: To estimate a major frame time.

PARAMETERS:
mfp_mainMjfPrevTimeArg : data_in
mfp_mainEstimatedArg : data_out
mfp_mainMjfTimeArg : data_out

RETURN VALUE:
Type = void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Parms	mfp_dataPeriod	lpsTimeStruct	I	data rate, mjf period
mfp_MjfQa	mfp_mjfQa	mfpMjfQaStruct		
	.mfp_tossedMjfCnt	integer	I	number of tossed vcdus

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:

End Unit "mfp_MainEstimateTime"
****

UNIT NAME: mfp_MainExtractTime

PURPOSE: To extract the major frame time (spacecraft time) from the 6 time code minor frames.

PARAMETERS:
mfp_mainNumBadDgsArg : data_out
mfp_mainBadTimeInfoArg : data_out
mfp_mainSpacecraftIdArg : data_out
mfp_mainMjfTcFramesArg : data_in
mfp_mainMjfTimeArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	The time code minor reduces with no errors.
LPS_FAILURE	The time code minor reduces, but with errors.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES: NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. On page 3-13 of the DFCB Volume 4, the time code minor frames indicate that certain groups of time information are N/A. The assumption is that the 40 bits for those groups are zeros and will be extracted by this routine.
2. If the time reduces properly, it will not be validated. If the time is wrong after BCH protection and the special time encoding protection, there is an extremely high probability that the spacecraft clock is off. MFPS is not prepared to handle a spacecraft clock error.

NOTES:
MFP_TOTAL_TCBYTES = total number of time code bytes possible

End Unit "mfp_MainExtractTime"
****

UNIT NAME: mfp_MainFillMissMjfs

PURPOSE: Generate a major frame with fill value for every missing major frame.

PARAMETERS:
mfp_mainMjfPrevTimeArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Indicates shared memory access was successful.
LPS_FAILURE	Indicates shared memory access failed.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_statusInfo	char	I	2-byte status info.
mfp_Global	mfp_subIntv	Structure	I	Subinterval info.
mfp_Parms	mfp_dataPeriod	real	I	Major frame data rate.
mfp_MjfQa	mfp_fullFillCnt	int	I	# missing mjfs

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: NONE

End Unit mfp_MainFillMissMjfs
****

```
UNIT NAME: mfp_MainFindMjfStart

PURPOSE: To find the start of a major frame by comparing the two VCDUs at a time,
         starting with the two VCDUs provided as arguments.

PARAMETERS:
mfp_mainPreAvInfoArg : data_inout
mfp_mainAvInfoArg : data_inout
RETURN_CODE : control_out

RETURN VALUE: RETURN_CODE
Type = RETURN_CODE

Value               Description
--------------      ------------------------------------------------------------
MFP_NO_CHANGE       No change in status of mfp_tossedVcduSet. (None of the
                    following values)
#
LPS_FAILURE         The call did not complete successfully
MFP_EOC             End of Contact detected
MFP_TOSSEDSET_FULL  The buffer holding pointers to 'tossed' VCDUs is full
MFP_SCANBIT_FLIP    The proper scan bit change has occured

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:
Source      Name                Type                 Use Description
----------- ------------------- ------------------   --- --------------------
mfp_Global  mfp_tossedVcduSet   mfpMjfVcduSetStruct  IO  VCDUs without a Mjf
mfp_Global  mfp_mjfVcduSet      mfpMjfVcduSetStruct  O   VCDUs in a Mjf
lps_shm_RDPMFP lps_annCadu      lpsAnnotatedCaduStruct I Ann Cadu Data

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:  rc is a local variable of type RETURN_CODE.

End Unit mfp_MainFindMjfStart
*************************************************************
```

```
UNIT NAME: mfp_MainGenSubIntv

PURPOSE: Generate the subinterval structure.

PARAMETERS:
mfp_mainDetSubInfoArg : data_in
mfp_mainMjfTimeArg : data_in

RETURN VALUE:
Type = void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
mfp_Global      mfp_subIntv         Structure       O   Subinterval Information

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: NONE

End Unit mfp_MainGenSubIntv
*****************************************************************
```

UNIT NAME: mfp_MainGetAvInfo

PURPOSE: To get pointer to the next Ann CADU in shared memory and to initialize fields of the Av Info argument.

PARAMETERS:
RETURN_CODE : control_out
mfp_mainAvInfoArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	The routine completed sucessfully
LPS_FAILURE	The routine did not complete sucessfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_nextAvInfo	mfpAvInfoStruct	O	Temp hold for a CADU
mfp_Global	mfp_nextAvInfoFlag	Boolean	IO	True if mfp_nextInfo is filled;
mfp_Global	mfp_shmAddList	mfpShmAddListStruct	IO	Structure holding Shm Block addresses
lps_shm_RDPMFP	lps_shmBlkRdpMfp	struct	I	Sh mem block
lps_shm_RDPMFP	lps_annCadu	lpsAnnotatedCaduStruct	IO	Ann Cadu Data

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
1. shmAcLoc is a static local to save CADU location from call to call. Initialized to 0 at startup.
2. shmBlkAdd is a static local to save the address of current sh mem Block. Initialized to a NULL value at startup.

End Unit mfp_MainGetAvInfo
****

```
UNIT NAME:mfp_MainGroupTC

PURPOSE: To group the time code information according to days, hours, minutes,
         seconds, and milliseconds.

PARAMETERS:
mfp_majorFrameTimeArg : data_in
mfp_mainMjfTimeArg : data_out

RETURN VALUE:
Type = void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES: NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:

End Unit "mfp_MainGroupTC"
*****************************************************************
```

```
UNIT NAME:mfp_MainIdentifyMjfSet

PURPOSE: To identify a set of VCDUs worth one major frame and save info about
         VCDUs missing from the set of  VCDUs that is processed. This module also
         creates a set of VCDUs which are 'tossed' because they could not be
         identified as belonging to a major frame.
PARAMETERS:
mfp_mainMjfSetStateArg : control_in
mfp_mainDetSubInfoArg : data_out
RETURN_CODE : control_out

RETURN VALUE:RETURN_CODE
Type = RETURN_CODE

Value              Description
--------------     ------------------------------------------------------------
LPS_FAILURE        The routine did not complete sucessfully
MFP_TOSSEDSET_FULL Exit since Tossed VCDU buffer is full (before Mjf synch VCDU)
MFP_EOC_NOMJF      EOC detected before Major Frame synch VCDU detected
MFP_EOC            EOC detected after Major Frame synch VCDU detected
MFP_NORMAL         Exit due to scanbit change after Mjf synch VCDU
MFP_END_FILLED     Fill CADUs at end of Mjf
MFP_VCID_CHANGE    Exit due to VCID change after Mjf synch VCDU
MFP_MJFSET_FULL    Exit due to overflow of Major frame VCDU buffer
MFP_DATA_BREAK     Exit due to loss of Mjf Synch

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:
Source      Name                Type                Use Description
----------- ----------------- --------------------- --- ------------------------
mfp_Global  mfp_tossedVcduSet mfpMjfVcduSetStruct   O   Set of tossed VCDUs
mfp_Global  mfp_mjfVcduSet    mfpMjfVcduSetStruct   O   VCDUs of a Mjf set

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

End Unit mfp_MainIdentifyMjfSet
*************************************************************
```

UNIT NAME: mfp_MainInit

PURPOSE: Performs all initialization for MFPS.

PARAMETERS:
mfp_mainThresFlagsArg : control_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Initialization was successful
LPS_FAILURE	Initialization failed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Parms	mfp_fillValue	char	I/O	1-byte fill value
mfp_Global	mfp_filledData	char	O	Preset fill area
mfp_Global	mfp_replacementCadu	String	O	Pseudo cadu full of fill
mfp_Global	mfp_pseudoAcPtr	Address	O	Ptr to pseudo cadu
mfp_Global	mfp_shmAddList	Struct	0	List of shm Addresses
mfp_Global	mfp_latestDataTime	lpsTimeStruct	0	Latest possible Mjf time for the contact period
mfp_Global	mfp_earliestDataTime	lpsTimeStruct	0	Earliest possible time for the contact period
mfp_SubQa	mfp_subQa	Struct	O	Initialize items to 0.
mfp_MjfQa	mfp_mjfQa	Struct	O	Initialize items to 0.
mfp_MjfQa	mfp_holdQa	Struct	O	Initialize items to 0.

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
MFPS is working on the same contact period as RDPS.

NOTES:

End Unit mfp_MainInit
****

```
UNIT NAME:mfp_MainInitVcduSets

PURPOSE: To initialize the VCDU Sets (both tossed and Mjf)
         before every VCDU collection process.

PARAMETERS:
mfp_mainMjfSetStateArg : control_in
mfp_mainMjfSetStartedArg : control_out

RETURN VALUE:
Type = Void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:
Source      Name                Type                 Use Description
----------- ----------------- --------------------- --- ------------------------
mfp_Global  mfp_tossedVcduSet mfpMjfVcduSetStruct   IO   VCDUs without a Mjf
mfp_Global  mfp_mjfVcduSet    mfpMjfVcduSetStruct   IO   VCDU of a Mjf
mfp_Global  mfp_mjfReuseNum   Integer               I   Num CADUs from end of
                                                        Mjf to be carried to next

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None
NOTES:

End Unit mfp_MainInitVcduSets
*************************************************************
```

```
UNIT NAME: mfp_MainMjfTime

PURPOSE:  Calculates the major frame time.

PARAMETERS:
mfp_mainMjfTcFramesArg : data_in
mfp_mainSeqEstCountArg : data_inout
RETURN_CODE : control_out
mfp_mainMjfTimeArg : data_inout

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    The major frame time is acceptable.
LPS_FAILURE    The major frame time is unacceptable.  Discard the major frame.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:
Source          Name                   Type            Use Description
--------------- --------------------- ------------- --- -----------------------
mfp_Parms       mfp_estSeqMax          integer        I  Maximum number of
                                                         estimated times to flow
                                                         through.
mfp_Qa          mfp_mjfQa              mfpMjfQaStruct
                 .mfp_timeCodeErrCnt   integer      I/O Time code error count

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:

End Unit mfp_MainMjfTime
*****************************************************************
```

UNIT NAME:mfp_mainPcdExtract

PURPOSE: Extracts the 4 Pcd bytes from specified VCDU set and puts the data in the shared memory for the PCDS subsystem.

PARAMETERS:
mfp_mainPcdInfoPtrArg : data_in
mfp_mainVcduSetArg : data_in
mfp_mainMjfSetEndArg : control_in

RETURN VALUE:
Type = void

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_mjfVcduSet	mfpMjfVcduSetStruct	I	Major frame VCDU set
mfp_Global	mfp_tossedVcduSet	mfpMjfVcduSetStruct	I	set of tossed VCDUs
lps_shm_MFPPCD	lps_shmBlkMfpPcd	lpsPcdInfoStruct	O	PCD info block
lps_shm_RDPMFP	lps_annCadu	lpsAnnotatedCaduStruct	I	Ann Cadu Data

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
NONE

NOTES:
None

End Unit "mfp_MainPcdExtract"
****

UNIT NAME: mfp_MainPcdStatusProc

PURPOSE: Calls the routines to extract Pcd/Status data from VCDUs.

PARAMETERS:
mfp_mainMjfValidArg : control_in
mfp_mainDetSubInfoArg : data_in
mfp_mainMjfSetEndArg : control_in
mfp_mainPcdStatSetFlgArg : control_out
RETURN_CODE : control_out

RETURN VALUE:
Type =RETURN CODE

Value	Description
LPS_SUCCESS	PCD/Status data extraction completed successfully
LPS_FAILURE	PCD/Status data extraction failed to complete successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_mjfVcduSet	mfpMjfVcduSetStruct	I	Major frame Vcdu set
mfp_Global	mfp_tossedVcduSet	mfpMjfVcduSetStruct	I	Set of tossed Vcdus
mfp_Global	mfp_subIntv	Sub_Intv	I	For Sub Int Id
lps_shm_MFPPCD	lps_shmBlkMfpPcd	lpsPcdInfoStruct	O	PCD info block

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit mfp_MainPcdStatusProc
****

UNIT NAME: mfp_MainQACalcSub

PURPOSE: Calculate quality and accounting information on subintv basis.

PARAMETERS:
mfp_mainNewSubIntvArg : control_in

RETURN VALUE:
Type = void

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_MjfQa	mfp_mjfQa	mfpMjfQaStruct	I	Major frame QA
mfp_MjfQa	mfp_holdQa	mfpMjfQaStruct	I	Major frame QA
mfp_SubQa	mfp_subQa	mfpSubQaStruct	O	Cum subintv QA

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mfp_MainQACalcSub
****

UNIT NAME: mfp_MainQASubGen

PURPOSE: Generate subinterval quality and accounting information.

PARAMETERS:
mfp_mainCalFileInfoArg : data_in
mfp_mainMscdFileInfoArg : data_in
mfp_mainThresFlagsArg : control_inout
mfp_mainNewSubIntvArg : control_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	The routine completed successfully
LPS_FAILURE	The routine did not complete successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_subIntv	struct	I	Sub Intv info

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. The BER count is not included in the QA statistics.
2. This rotine is called by mfp_Main only when
   a. EOC occurs or
   b. A Mjf is received.

NOTES:
None

End Unit mfp_MainQASubGen
****

UNIT NAME: mfp_MainQAThrTest

PURPOSE: Threshold test for quality and accounting information summed over a subinterval.

PARAMETERS:
mfp_mainThresFlagsArg : control_inout

RETURN VALUE:
Type = void

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_SubQa	mfp_subQa	mfpSubQaStruct	I	Subintv QA data
mfp_Thres	mfp_thres	Struct	I	MFP threshold values

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mfp_MainQAThrTest
****

UNIT NAME: mfp_MainQAZeroData

PURPOSE: Reinitialize mfp_MjfQa and mfp_SubQa.

PARAMETERS:
mfp_mainThresFlagsArg : control_out
mfp_mainNewSubIntvArg : control_in

RETURN VALUE:
Type = void

Value	Description

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_MjfQa	mfp_mjfQa	mfpMjfQaStruct	I/O	Enters with current MJF QA and exits empty
mfp_MjfQa	mfp_holdQa	mfpMjfQaStruct	I/O	Enters with current hold QA and exits empty
mfp_SubQa	mfp_subQa	mfpSubQaStruct	O	Start of new subintv QA

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit mfp_MainQAZeroData
****

```
UNIT NAME: mfp_MainStatusExtract

PURPOSE:To extract the ETM+ status data from a valid major frame.

PARAMETERS:
mfp_mainPcdStatSetFlgArg : control_out

RETURN VALUE:
Type =VOID

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
NONE

EXTERNAL VARIABLES:
Source             Name                Type            Use Description
----------     ------------------- ---------------     --- ----------------------
-
mfp_Global     mfp_StatusInfo      mfpStatusInfoStruct O   holds ETM+ status data
lps_shm_RDPMFP lps_annCadu         lpsAnnotatedCaduStruct I Ann Cadu Data

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
NONE
NOTES:
NONE

End Unit mfp_MainStatusExtract
*************************************************************
```

UNIT NAME: mfp_MainTCMnfSync

PURPOSE: To extract and check for the sync or end of time code patterns in minor frames 1 and 6.

PARAMETERS:
mfp_mainDataWordArg : data_in
mfp_mainTcTypeArg : data_in
mfp_mainTCMnfArg : data_in
lpsBooleanType : control_out

RETURN VALUE:
Type = lpsBooleanType

Value	Description
LPS_TRUE	The minor frame matches the pattern.
LPS_FALSE	The minor frame does not match the pattern.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Parms	mfp_tcPattern	char	I	time code pattern

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:

End Unit "mfp_MainTCMnfSync"
****

UNIT NAME: mfp_MainValidateMjf

PURPOSE: This module calls routines to check VCDUs in the shared memory to identify start and end of a major frame and to compute major frame time and quality and accounting data for the major frame.

PARAMETERS:
mfp_mainSeqEstCountArg : data_inout
mfp_mainMjfValidArg : control_out
mfp_mainBandsArg : data_out
mfp_mainMjfTimeArg : data_out
mfp_mainCalMscdArg : data_out
RETURN_CODE : control_out
mfp_mainDetSubInfoArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_FAILURE	The call did not complete successfully
MFP_TOSSEDSET_FULL	No Major Frame; Tossed VCDU set became full
MFP_EOC_NOMJF	No Major Frame; EOC detected during 'Tossing' of VCDUs
MFP_EOC	EOC detected after Major Frame synch VCDU detected
MFP_NORMAL	Exit due to scanbit change after Mjf synch VCDU
MFP_END_FILLED	Fill CADUs at end of Mjf
MFP_VCID_CHANGE	Exit due to VCID change after Mjf synch VCDU
MFP_MJFSET_FULL	Exit due to overflow of Major frame VCDU buffer
MFP_DATA_BREAK	Exit due to loss of Mjf Synch

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_MjfQa	mfp_mjfQa	mfpMjfQaStruct	IO	Totals on Mjf basis
mfp_MjfQa	mfp_holdQa	mfpMjfQaStruct	IO	Totals for tossed Sets
mfp_Global	mfp_mjfReuseNum	Integer	O	Num CADUs from end of Mjf to be carried to next Mjf
mfp_Global	mfp_mjfVcduSet	mfpMjfVcduSetStruct	I	Major frame VCDU set
mfp_Global	mfp_tossedVcduSet	mfpMjfVcduSetStruct	I	set of tossed VCDUs

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
1. mjfSetState is a static local

End Unit mfp_MainVaidateMjf
****

UNIT NAME:mfp_MainValidateTime

PURPOSE: To validate the major frame time.

PARAMETERS:
mfp_mainCheckTypeArg : data_in
lpsBooleanType : control_out
mfp_mainNumBadDgsArg : data_in
mfp_mainBadTimeInfoArg : data_in
mfp_mainEstimatedArg : data_in
mfp_mainMjfTimeArg : data_in

RETURN VALUE:
Type = lpsBooleanType

Value	Description
LPS_TRUE	Actual time is equivalent to the expected time
LPS_FALSE	Actual time is not within the tolerance range of the expected time

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Parms	mfp_timeEqualTol	lpsTimeStruct	I	Range in which to consider two times equivalent
mfp_Global	mfp_earliestDataTime	lpsTimeStruct	I	Earliest possible data time
	mfp_lastestDataTime	lpsTimeStruct	I	Latest possible data time

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:

End Unit "mfp_MainValidateTime"
****

```
UNIT NAME: mfp_MainValVcidChg

PURPOSE:  To validate the VCID change, indicated by RDPS via annotation.

PARAMETERS:
mfp_mainPreAvInfoArg : data_inout
mfp_mainValidFlagArg : control_out
mfp_mainVcduSetArg : data_in
mfp_mainAvInfoArg : data_inout

RETURN VALUE: RETURN_CODE
Type = void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. If one or both VCIDs being compared are not known, then it is
   assumed that the VCID has not been changed.

NOTES:
NONE

End Unit mfp_MainValVcidChg
*****************************************************************
```

```
UNIT NAME: mfp_MainVerifySpacecraftId

PURPOSE: Verifies that the spacecraft Id extracted from the time code is
         correct.

PARAMETERS:
lpsBooleanType : control_out
mfp_mainSpacecraftIdArg : data_in

RETURN VALUE:
Type = lpsBooleanType

Value          Description
-------------- ------------------------------------------------------------------
LPS_TRUE       The spacecraft Id extracted is correct.
LPS_FALSE      The spacecraft Id extracted is not correct.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
mfp_Parms       mfp_spacecraftId     char            I   spacecraft Id

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:

End Unit mfp_MainVerifySpacecraftId
*****************************************************************
```

UNIT NAME: mfp_MscdL0rExtract

PURPOSE: Extract the mirror scan correction data.
PARAMETERS:
lpsInt32Type : data_out
mfp_mscdNewSubIntvArg : data_in
mfp_mscdDetSubInfoArg : data_in
mfp_mscdFpArg : data_in
mfp_mscdCalMscdArg : data_in
mfp_mscdMjfTimeArg : data_in

RETURN VALUE:
Type = lpsInt32Type

Value	Description
mfp_mscdScanDirArg	The scan direction of type integer.
mfp_mscdCalStartArg	A structure with fields of type integer pointing to the start of the calibration data.
LPS_SUCCESS	The routine executed successfully.
LPS_FAILURE	The routine executed unsuccessfully.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
NONE		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_MjfQa	mfp_caduSyncErrCnt	integer	I	number of CADU sync Err
mfp_MjfQa	mfp_eolErrFlag	boolean	I	end of line error
mfp_MjfQa	mfp_partFillCnt	integer	I	number of frames filled
mfp_MjfQa	mfp_timeCodeErrCnt	integer	I	time code error

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

End Unit mfp_MscdL0rExtract
****

UNIT NAME: mfp_MscdWriteFile

PURPOSE: Write Mscd Data and File header information to the MSCD File.

PARAMETERS:
mfp_mscdDataArg : data_in
mfp_mscdNewSubIntvArg : data_in
mfp_mscdMjfTimeArg : data_in
mfp_mscdFpArg : data_in
mfp_mscdDetSubInfoArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Successfully writes data to the Mscd file
LPS_FAILURE	Unsuccessfull data write to the Mscd file

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
MSCD_File	O	Stores the Mscd data.

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
mfp_Global	mfp_mscdFileName	char	I	Mscd File name
mfp_Global	mfp_subIntv	struct	I	Subinterval information
mfp_Global	mfp_statusInfo	struct	I	Status information
mfp_Global	mfp_spacecraftId	integer	I	spacecraft id

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
NONE

NOTES:
NONE

End Unit "mfp_MscdWriteFile"
****

```
* UNIT NAME:   mfp_VerifyMjf
*
* PURPOSE:   Verify that critical Major-frame info is present in
*                the VCDUset.
*
* UNIT TYPE:   (int) function
*

PARAMETERS:
mfp_mainMjfSetEndArg : control_in
mfp_mainMjfReuseNumArg : data_out
mfp_mainMjfTcFramesArg : data_out
RETURN_CODE : control_out
mfp_mainBandsArg : data_out
mfp_mainCalMscdArg : data_out

* RETURN VALUES:
*    MFP_VALID   =  good MajFrm or recoverable_error(s):
*                      fill/partial-fill for sync,
*                      fill/partial time code
*                      partial EOL
*                      fill/partial-fill/error for next sync
*    MFP_EOL_ERR = EOL not found, may have other recoverable errs.
*    MFP_TOSSED  = non-recoverable error(s) - delete MajFrm
*                      mfp_mainSetEndArg == MF_MJFSET_FULL
*                      sync_index not in VCDUset
*                      invalid offset for computing sync location
*                      sync location contains other data, not sync
*
* ARGUMENTS:
*    NAME                TYPE   use DESCRIPTION
*    ---------------------------------------------------------------------
*    mfp_mainSetEndArg    int    I  end condition for VCDUset:
*                                       MFP_MJFSET_FULL -
*                                         the impossible has happened:
*                                         the VCDUset has filled up.
*                                         VCDUcnt,mfc,offset all continue
*                                         incrementing correctly as within
*                                         the major frame, without a scan
*                                         bit change and without a break
*                                         as from the mfc,offset from a
*                                         new major frame or rt/pb switch.
*                                       MFP_NORMAL -
*                                         the last CADU in the array has a
*                                         scan-bit change, a sequential
*                                         VCDUcnt, and the mfc,offset does
*                                         not match the sequence for the
*                                         major frame.
```

```
*                                       MFP_END_FILLED -
*                                         the last CADU in the array has a
*                                         scan-bit change, a jump in the
*                                         VCDUcnt from the previous
*                                         non-fill CADU, and the mfc,offset
*                                         does not match the sequence for
*                                         the major frame.  Fill CADU(s)
*                                         have been inserted between the
*                                         last CADU and the preceding data
*                                         CADU.
*                                       MFP_DATA_BREAK -
*                                         the last CADU in the array has a
*                                         jump in the VCDUcnt, and the
*                                         VCDUcnt,mfc,offset does not match
*                                         the sequence for the major frame.
*                                         No fill has been inserted.
*                                       MFP_VCID_CHANGE -
*                                         the last CADU in the array has a
*                                         different VCID than the
*                                         preceding CADUs.
*    mfp_mainBandsArg    struct* O  values for band info extraction:
*      .mfp_syncLoc      struct       sync frame info:
*         .mfp_vcduIndex  int           index to VCDUset array for CADU
*         .mfp_byteLoc    int           byte offset in CADU data area
*      .mfp_sceneEndMfc   int         mfc for last mf in scene
*      .mfp_band6EndMfc   int         mfc for last full mf in Mjf
*                                        (= end of band 6 data)
*    mfp_mainMjfTcFramesArg
*                        struct* O  time code info:
*      .mfp_tcMnfs        uchar       buffer for 6 minor frames of GMT
*      .mfp_tcCompFlag    int         completeness indicator for GMT:
*                                       0=missing, 1=complete,
*                                       2=partial(start), 3=partial(end)
*      .mfp_totalNumBytes int         # bytes of data in buffer
*    mfp_mainCalMscdArg  struct* O  cal/mscd extraction info:
*      .mfp_calMscdLoc   struct       scan line data info:
*         .mfp_vcduIndex  int           index to VCDUset array for CADU
*         .mfp_byteLoc    int           byte offset in CADU data area
*      .mfp_lastMnfLoc   struct       last mf (full/partial) info:
*         .mfp_vcduIndex  int           index to VCDUset array for CADU
*         .mfp_byteLoc    int           byte offset in CADU data area
*      .mfp_lastMnfSize   int         # bytes present from last mf
*    mfp_mainMjfReuseNumArg
*                         int*   O  CADU re-use count for next MajFrm
*                                     0-N = re-use last # data/fill CADUs
*                                     -1  = re-use last (data) CADU only,
*                                             insert fill CADUs as needed
*
* EXTERNAL VARIABLES:
*    SOURCE    NAME         TYPE  use  DESCRIPTION
*    ----------------------------------------------------------------------
*    "mfp_Global.h"
*         mfp_mjfVcduSet    struct
*           .mfp_vcduCnt     int   I   # of CADUs in array
*           .mfp_avInfo     struct I   array of CADU info
*
```

```
* EXTERNAL FILE/RECORD REFERENCES:
*    NAME                USE      DESCRIPTION
*    ----------------------------------------------------------------------
*    <stdio.h>        compile    Define I/O calls, structures
*    <stdlib.h>       compile    define system routine calls
*    "lps_?????.h"    compile    define LPS structures, #define's, etc.
*    "mfp_?????.h"    compile    define MFP structures, #define's, etc.
*    "mfp_Global.h"   compile    define list of CADUs in major frame
*
* EXTERNAL FUNCTIONS:
*    mfp_FindMjfSync   - find sync minor frame in the current/next
*                          Major frame
*    mfp_FindMjfEol    - find the EOL in the Major frame
*    mfp_ExtractMnf    - extract N minor frames from the CADUs
*
* Non_Portable Code:
*    1. Description:  machine/compiler-dependent automatic padding
*          at end of annotated CADU structure as defined in header
*          file " ".  Each annotated CADU must start on an 8-byte
*          boundary.  Since the annotated CADU as defined is not a
*          multiple of 8 bytes, the SGI compiler (most likely)
*          automatically inserts 3 bytes of pad before beginning the
*          next annotated CADU.  If LPS is transported to a single
*          4-byte or 8-byte word machine, the software should still
*          work.  If transported to a single byte-oriented machine,
*          the pad will disappear, producing smaller records which
*          may affect trouble files or other file access, but the
*          general processing should still work.  If LPS is
*          distributed to a mix of byte and 4/8-byte machines, data
*          passed from one machine to the next will be subject to 2
*          different physical interpretations of the same code, and
*          will fail.
*       Reason for use:  management re-write of the data structure
*          definition excluding explicit pad
*       Suggested porting procedure:  re-insert the explicit pad
*
* Assumptions:
*    1. The DFCB, with corrections per the Gary Trup memo, defines
*         the CADU and major frame data.  The Jackson & Tull memo of
*         June 7,1995 is incorrect and not usable.
*    2. The implementer is expected to be familiar with the relevant
*         satellite data structures, the current mfp environment,
*         and the routines called by this function.
*    3. The list of CADUs for the major frame must start with a
*         scan-direction flag change between the 1st 2 CADUs.  Either
*         or both of these may be MFP-generated fill, and will be
*         flagged as such by the generating routine.
*    4. The list of CADUs for the major frame must end with a data
*         (non-fill) CADU.  The following sub-conditions apply:
*         normal- the last CADU and the previous non-fill CADU
*                 contain a scan-direction change.  If any data
*                 CADUs are missing between them, fill CADUs will
*                 have been inserted by the calling routine.
*         other(EOC, VCID change, discontinuity) -
*                The end of the major frame will NOT be filled.
*                The last CADU is NOT part of the current major frame.
*    5. VCIDs (except for the last CADU) are expected to be the same.
*    6. mf_count, and byte_offset in the CADU are expected to be
```

```
*         correct for sync determination.  The sync determination
*         routines do some range checking to detect bad values, but
*         errors within range will still pass thru to produce errors
*         in locating the sync frame.  No further checking is done
*         for locating the EOL or extracting the time frames.
*
*
End Unit  mfp_VerifyMjf
*********************************************************
```

```
UNIT NAME:
        pcd_db_GetFirstWrsScene

PURPOSE:
        Retrieve the nearest WRS nominal scene center from database

PARAMETERS:
pcd_mainVelArg : data_in
RETURN_CODE : control_out
pcd_mainFirstSceneArg : data_out
pcd_mainLongArg : data_in
pcd_mainLatArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    The nearest WRS scene center was successfully retrieved
LPS_FAILURE    The nearest WRS scene center was not successfully retrieved

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Valid_WRS_Parms            I   WRS scene centers table

EXTERNAL VARIABLES: NONE
Source          Name                   Type            Use Description
--------------- --------------------   --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:
1. When the satellite travels southward, the z component of the velocity is
   negative; when the satellite travels northward, the z component of the
   velocity is positive.

End Unit pcd_db_GetFirstWrsScene
*****************************************************************
```

```
UNIT NAME:
        pcd_db_GetNextWrsScene

PURPOSE:
        Retrieve the next WRS scene center from database

PARAMETERS:
RETURN_CODE : control_out
pcd_mainNextWrsSceneArg : data_out
pcd_mainNomSceneArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    The next WRS scene center was successfully retrieved
LPS_FAILURE    The next WRS scene center was not successfully retrieved

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Valid_WRS_Parms            I   WRS scene centers table

EXTERNAL VARIABLES: NONE
Source          Name                   Type            Use Description
--------------- -------------------    --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: NONE

End Unit pcd_db_GetNextWrsScene
*****************************************************************
```

```
UNIT NAME:
        pcd_db_GetPrevWrsScene

PURPOSE:
        Retrieve the previous WRS scene center from database

PARAMETERS:
RETURN_CODE : control_out
pcd_mainPrevWrsSceneArg : data_out
pcd_mainNomSceneArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    The previous WRS scene center was successfully retrieved
LPS_FAILURE    The previous WRS scene center was not successfully retrieved

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Valid_WRS_Parms            I   WRS scene centers table

EXTERNAL VARIABLES: NONE
Source          Name                   Type            Use Description
--------------- -------------------    --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: NONE

End Unit pcd_db_GetPrevWrsScene
*****************************************************************
```

```
UNIT NAME:
        pcd_db_GetValidParms

PURPOSE:
        Retrieve the Valid PCD Parameters from the database.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS     The valid parameters were retrieved successfully.
LPS_FAILURE     The valid parameters were not retrieved.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Valid_PCD_Parms            I  Valid PCD processing parameters.

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
pcd_Global      pcd_validParms      pcdValidParmStruct
                                                     O  The valid parameters
                                                        used for PCD processing

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
NONE.

NOTES:

End Unit pcd_db_GetValidParms
*****************************************************************
```

```
UNIT NAME:
        pcd_db_GetValidSceneParms

PURPOSE:
        Retrieve the valid scene parameters from the database.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS     The PCD scene parameters were retrieved successfully.
LPS_FAILURE     The PCD scene parameters were not retrieved successfully.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -------------------------------------------------
Valid_Scene_Parms          I  Scene parameters needed for scene identification.

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
pcd_Global      pcd_validSceneParms pcdValidSceneParmsStruct
                                                     O  The valid scene
parameters
                                                        used during PCD
processing

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
NONE.

NOTES:
NONE.

End Unit pcd_db_GetValidSceneParms
*****************************************************************
```

```
UNIT NAME:
        pcd_db_GetValidThres

PURPOSE:
        Retrieves the valid PCD thresholds from the database.

PARAMETERS:

RETURN VALUE:
Type =  Void

Value          Description
-------------- ------------------------------------------------------------------
NONE.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Valid_PCD_Thres            I   The valid PCDS thresholds.

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
pcd_Global      pcd_validThres      pcdValidThresStruct
                                                     O  The valid threshold
                                                        structure.

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
Default values for each of the elements in pcd_validThres have been defined and
will be used when the actual thresholds cannot be retrieved from the LPS data-
base.

NOTES:
NONE.

End Unit pcd_db_GetValidThres
*************************************************************
```

UNIT NAME: pcd_db_ReportSceneInfo

PURPOSE: Reports each scene description to the database.

PARAMETERS:
pcd_mainSceneArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	The scene description was successfully entered into the database
LPS_FAILURE	The scene description was not entered into the database successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
PCD_Scene_Acct	O	Database table containing the scene accounting on a contact period basis

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
pcd_Global	pcd_contactSeqId	lpsUInt32Type	I	Contact sequence ID
pcd_Global	pcd_fileVersionNum	lpsUInt32Type	I	File version number

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit pcd_db_ReportSceneInfo
****

```
UNIT NAME: pcd_db_StoreAcctInfo

PURPOSE: Stores the PCD quality and accounting on a subinterval basis.

PARAMETERS:
pcd_accSubIntvIdArg : data_in
pcd_accSubIntvAcctArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    The accounting was inserted successfully
LPS_FAILURE    The accounting was not inserted successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -------------------------------------------------
PCD_Acct                   O  Contains the PCD metadata accounting information
                              on a subinterval basis.

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit pcd_db_StoreAcctInfo
*****************************************************************
```

```
UNIT NAME:
        pcd_db_StoreBandsPresent

PURPOSE:
       Stores the bands present flags in the LPS database.

PARAMETERS:
RETURN_CODE : control_out
pcd_storeBandInfoArg : data_in
pcd_storeBandInfoCountArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ----------------------------------------------------------------
LPS_SUCCESS    The bands present info was stored in database successfully
LPS_FAILURE    The bands present info was not stored in the database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Bands_Present               O  Database table for storing the bands present
                               flags for each PCD Cycle.

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
pcd_Global      pcd_contactSeqId     lpsUInt32Type    I  The contact sequence
                                                           identifier
pcd_Global      pcd_fileVersionNum   lpsUInt32Type    I  The file version number

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit pcd_db_StoreBandsPresent
*****************************************************************
```

```
UNIT NAME:
        pcd_LagrangeInt

PURPOSE:
        Perform the N-point Lagrange interpolation

PARAMETERS:
pcd_xArg : data_in
pcd_fxArg : data_in
pcd_xinArg : data_in
pcd_numArg : data_in
pcd_foutArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE
Name                       Use Description
------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES: NONE
Source          Name                 Type            Use Description
--------------- ------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:
 1. The Lagrange interpolation algorithm is listed as follows:

    pcd_foutArg = L(1) * pcd_fxArg(1) + L(2) * pcd_fxArg(2) + ... +
                  L(pcd_numArg) * pcd_fxArg(pcd_numArg)
       where L(I) = (PROD(pcd_xinArg-pcd_xArg(J)))/
                    (PROD(pcd_xArg(I)-pcd_xArg(J)))
             J = 1 thru pcd_numArg and J is not equal to I

End Unit pcd_LagrangeInt
*************************************************************
```

UNIT NAME: pcd_Main

PURPOSE: Controls the Level 0R PCD data processing for a contact period.

PARAMETERS:
RETURN_CODE : control_out
pcd_mainFileVersionArg : data_in
pcd_mainContactSeqIdArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Successful completion of PCD processing
LPS_FAILURE	Failure in completing the PCD processing

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
None		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
pcd_Global	pcd_eocFlag	lpsBooleanType	I	End of contact period indicator
pcd_Global	pcd_procReturnCode	lpsStatusType	I	PCDS processing status code
pcd_Global	pcd_contactSeqId	lpsUInt32Type	I	Contact sequence id
pcd_Global	pcd_fileVersionNum	lpsUInt32Type	I	L0R File version number
pcd_Global	pcd_cycleList	pcdCycleStruct	I	List of PCD Cycles
pcd_Global	pcd_subIntvList	pcdSubIntvStruct	I/O	Subinterval information within contact period
pcd_Global	pcd_sceneList	lpsSceneDescrStructPtr	I	Pointer to a linked list of scene center descriptions

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
1) If a break occurs within the PCD data, make certain that pcd_MainDetermineScenes is not called if the broken section of data contains less than 6 major frames worth of PCD data
2) pcd_db_ReportSceneInfo removes the nodes from the linked list as they are inserted into the database..

End Unit pcd_Main:
****

```
UNIT NAME: pcd_MainAcceptMinorFrames

PURPOSE: Create PCD minor frames linked list by adding valid minor frames to the
          tail of the list.
PARAMETERS:
pcd_mainPositionInBuffArg : data_in
pcd_mainSlippageArg : data_in
pcd_mainInvertedMinorFrameArg : data_in
pcd_mainNumFlippedBitsArg : data_in
pcd_mainAlignedMinorFramesArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_FS_SUCCESS     The PCD Cycles were successfully constructed
LPS_FS_FAILURE     The PCD Cycles were not successfully constructed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type                Use Description
--------------- -------------------- ---------------     --- ----------------------
-
pcd_Global      pcd_minorFrames      pcdMinorFrameStruct O  Linked list of PCD
minor
                                                            frames
NOTES:
1) Only minimal data quality verification. Additional validation may be
   required.
2) Quality accounting information will need to be collected at the minor
   frame level. The information to be collected is yet to be determined,
   but it will need to be supplied either by the frame sync routine or
   by global variables.

End Unit pcd_MainAcceptMinorFrames:
*****************************************************************
```

```
UNIT NAME:
        pcd_MainAdjustTime

PURPOSE:
        Determine the ephemeris and attitude time.

PARAMETERS:
pcd_mainMjfTimeArg : data_out
pcd_mainScTimeArg : data_in

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
pcd_Global      PCD_MJF_TIMEUPDATE   float           I   pcd major frame time
                                                         period

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:
 1. The current major frame is incremented by 4.096 seconds from the previous
one.

End Unit pcd_MainAdjustTime
*****************************************************************
```

```
UNIT NAME:
        pcd_MainAttitudeInt

PURPOSE:
        Interpolate the attitude data point using the Lagrange interpolation
method

PARAMETERS:
pcd_mainAttTimeArg : data_in
pcd_mainAttDataArg : data_in
pcd_mainTimeArg : data_in
pcd_mainNumPtrArg : data_in
pcd_mainIntAttArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE
Name                       Use Description
-------------------------- --- -----------------------------------------------

EXTERNAL VARIABLES: NONE
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: NONE

End Unit pcd_MainAttitudeInt
*****************************************************************
```

```
UNIT NAME:  pcd_MainBuildCycle

PURPOSE:  Drives the generation of the PCD Cycle

PARAMETERS:
pcd_mainDataWordArg : data_in
RETURN_CODE : control_out

RETURN VALUE:  Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_FAILURE     The PCD Cycles were not constructed successfully
LPS_SUCCESS     The PCD Cycles were constructed successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -----------------------------------------------
NONE.

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
pcd_Global      pcd_subIntvChangedFlag
                                    lpsBooleanType   I  ETM+ sub-interval change
                                                       indicator.
pcd_Global      pcd_minorFrames     pcdMinorFrameStruct
                                                     O  Pointer to the PCD minor
                                                        frames

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:  NONE

NOTES:
None

End Unit pcd_MainBuildCycle
*****************************************************************
```

```
UNIT NAME:  pcd_MainBuildMajorFrames

PURPOSE:  Generate a linked list of PCD major frames from the passed list of
          PCD minor frames.
PARAMETERS:
pcd_mainMinorFrameArg : data_in
pcd_mainMajorFrameArg : data_out
RETURN_CODE : control_out

RETURN VALUE:  Type = RETURN_CODE
Value          Description
-------------- ------------------------------------------------------------------
LPS_FAILURE     The PCD major frames were built successfully
LPS_SUCCESS     The PCD major frames were not built successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
pcd_Main        pcd_eocFlag          lpsBooleanType   I  The end of contact
                                                          period indicator
pcd_Main        pcd_subIntvChangedFlag
                                     lpsBooleanType   I The end of an ETM+ sub-
                                                          interval indicator

NOTES:  Current algorithm is very basic and dependent solely on the major and
         minor frame counters -- may want to perform time based correction at
         a later date.
        Need to come up with an algorithm to deal with corrupted minor frames --
         may need to access statistics from minor frame generation.
        Major frame ID within minor frame is only available for major frames 1-3.
         -- Need to determine what major frame a minor frame belongs to. In the
         case of major frame zero, it will be necessary to check the time field
         that is in place of the major frame ID (Should this be done at major
frame
         generation instead of at PCD cycle generation?).

End Unit pcd_MainBuildMajorFrames:
*************************************************************
```

```
UNIT NAME: pcd_BuildMinorFrames

PURPOSE: Builds the PCD Minor Frames using the PCD Information Words.

PARAMETERS:
pcd_mainProcStatusArg : data_in
pcd_mainDataWordArg : data_in
RETURN_CODE : control_out

RETURN VALUE: Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS     The minor frames were built successfully
LPS_FAILURE     The minor frames were not built successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -----------------------------------------------
NONE.

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
pcd_Global      pcd_eocFlag         lpsBooleanType   I  Flag to indicate the end
                                                          of a contact period
pcd_Global      pcd_minorFrames     pcdMinorFrameStruct
                                                    I/O Pointer to the list of
                                                          identified minor frames.

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:
   fs_params and fs_stats are local variables which are initialized
   when declared.

End Unit pcd_BuildMinorFrames
*****************************************************************
```

```
UNIT NAME:  pcd_MainCheckDataPoints

PURPOSE:  Determines the quality of the ephemeris and attitude data points
          for each PCD major Frame in the current PCD Cycle.

PARAMETERS:
pcd_mainPcdCycleArg : data_inout

RETURN VALUE:  Type = void

Value          Description
-------------- -------------------------------------------------------------
None

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ----------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
pcd_Global      pcd_validThres      pcdValidThresStruct
                                                     I  Contains thresholds for
                                                         valid ephemeris and
                                                         attitude data points

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit pcd_MainCheckDataPoints:
****************************************************************
```

UNIT NAME: pcd_MainCleanUp

PURPOSE: Cleans-up a contact period processing session for the PCDS.

PARAMETERS:

RETURN VALUE:
Type = void

Value	Description
None	

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
None		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
pcd_Global	pcd_procReturnCode	lpsStatusType	I	PCDS processing status
pcd_Global	pcd_cycleList	pcdCycleStructPtr	I	Pointer to PCD Cycle linked list

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
None

End Unit pcd_MainCleanUp
****

```
UNIT NAME:  pcd_MainComputeHds

PURPOSE:    Computes the horizontal display shift (HDS) between the nominal scene
            coordinates and the actual scene coordinates
PARAMETERS:
pcd_mainHdsArg : data_out
pcd_mainActSceneArg : data_in
pcd_mainNomSceneArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    The HDS was successfully computed
LPS_FAILURE    The HDS was not successfully computed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
pcd_Global      pcd_validSceneParms pcdValidSceneParmStruct
                                                     I   IAS data items needed
                                                         for WRS scene
                                                         identifications
NOTES:
 1. The HDS is defined as the perpendicular distance of the actual scene center
    from the nominal scene center track.  The positive HDS means that the actual
    scene center is on the right side of the track; while the negative HDS means
    that the actual scene center is on the left side of the track.
 2. The math algorithm is documented in LPS WRS Horizontal Display Shift
    Algorithm dated August 18, 1995.

End Unit pcd_MainComputeHds
*****************************************************************
```

```
UNIT NAME:
        pcd_MainComputeLat

PURPOSE:
        Compute the geodetic latitude of the look point

PARAMETERS:
pcd_mainPosArg : data_in
pcd_mainDeclnArg : data_in
pcd_mainRightAsnArg : data_in
pcd_mainLosGciArg : data_in
pcd_mainLatArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
pcd_Global      pcd_validSceneParms pcdValidSceneParmStruct
                                                     I   IAS data items needed
                                                         for WRS scene
identifications

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: The math algorithm is documented in algorithm #2 section of WRS Scene
       Identification Algorithms dated July 31, 1995.

End Unit pcd_MainComputeLat
*****************************************************************
```

```
UNIT NAME:
        pcd_MainComputePosition

PURPOSE:
        Compute the latitude and longitude for each of the PCD Major Frames.

PARAMETERS:
pcd_mainSceneParmArg : data_in
pcd_mainPositionArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type =RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS     The lat and long were successfully calculated
LPS_FAILLURE    The lat and long were not successfully calculated.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
pcd_Global      pcd_validSceneParms pcdValidSceneParmStruct
                                                     I   The valid scene parms
                                                         needed to compute the
                                                         longitude and latitude
                                                         of the point of view

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: NONE

End Unit pcd_MainComputePosition
*****************************************************************
```

```
UNIT NAME: pcd_MainConstructCycles

PURPOSE: Constructs the PCD Cycles using the PCD Major Frames.

PARAMETERS:
pcd_mainMajorFrameArg : data_in
RETURN_CODE : control_out

RETURN VALUE: Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS     The PCD Cycles were successfully constructed
LPS_FAILURE     The PCD Cycles were not successfully constructed

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -----------------------------------------------
NONE.

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
pcd_Global      pcd_subIntvChangedFlag
                                    lpsBooleanType   I  Flag that indicates
                                                         the change of an ETM+
                                                         sub-interval.
pcd_Global      pcd_eocFlag         lpsBooleanType   I  Flag that indicates
                                                         the end of a contact
                                                         period.
pcd_Global      pcd_cycleList       pcdCycleStruct  I/O Pointer to list of
                                                        PCD cycles

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:  NONE

NOTES: NONE

End Unit pcd_MainConstructCycles:
****************************************************************
```

```
UNIT NAME:  pcd_MainCreatePcdFile

PURPOSE: Creates the PCD file and accounting on a sub-interval basis.
PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:  Type = RETURN_CODE

Value          Description
-------------- ----------------------------------------------------------------
LPS_SUCCESS     A PCD File was created successfully
LPS_FAILURE     A PCD File was not created successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
PCD_File                    O  The PCD L0R File.
PCD_Trouble_File            O  The PCDS trouble file.

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
pcd_Global      pcd_eocFlag          lpsBooleanType   I  Flag that indicates the
                                                           end of contact period
pcd_Global      pcd_subIntvList      pcdSubIntvStruct I  Structure of PCD
subinterval
                                                           information
pcd_Global      pcd_cycleList        pcdCycleStruct   I  Pointer to a list of PCD
                                                           cycles
ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
   Any PCD cycles with a time stamp that is earlier than the first ETM+ Major
   Frame Time of a contact period will be stored in the PCD Trouble File.
   Any PCD cycles with a time stamp that is later than the last ETM+ Major Frame
   Time of a contact period will be stored in the PCD Trouble File.

NOTES:
  - pcd_subIntvList is a structure that contains records with the Q&A and
    file descriptor for each subinterval within the contact period.
  - Need to verify that PCD accounting structure contains adequate information.
  - HDF file format is yet to be implemented.

End Unit pcd_MainCreatePcdFile:
****************************************************************
```

```
UNIT NAME:
     pcd_MainDetermineMissingWords

PURPOSE:
     Determines the number of PCD words missing based upon supplied counters.

PARAMETERS:
pcd_mainOffsetArg : data_in
pcd_mainMissingDataWordCntArg : data_out
pcd_mainStartPosArg : data_in

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
NONE

NOTES:
NONE

End Unit pcd_MainDetermineMissingWords
*****************************************************************
```

```
UNIT NAME: pcd_MainDeterminOrbitNum

PURPOSE: Determines the orbit number for the sub-interval.

PARAMETERS:
pcd_mainPcdStartTimeArg : data_in
pcd_mainOrbitNumArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
None

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ---------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
pcd_Global      pcd_validSceneParms  pcdValidSceneParmStruct
                                                     I   The scene parameters
                                                         needed to determine orbit
                                                         number
ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) It is assumed that the first orbit is considered orbit number one.
2) The calculated orbit number will be rounded to the nearest whole value.
3) A default value will be used as the orbit number if the orbit number
   cannot be determined due to an invalid first major frame time
4) The lps_TimeDiff routine subtracts lps_time2Arg from lps_time1Arg.
   and returns the difference.
5) The lps_TimeDiff routine accepts the time as an integer type

NOTES:
1) The firstOrbitDetermination is a flag that indicates the first attempt
   to determine the orbit number within a contact period.
2) firstOrbitDetermination is a static local variable that is initialized to
   LPS_TRUE in the declaration section.
3) missionStart and orbitTime are static local variables

End Unit pcd_MainDetermineOrbitNum
*****************************************************************
```

```
UNIT NAME: pcd_MainDeterminePcdWord

PURPOSE: Extracts the information words from the PCD Byte stream.

PARAMETERS:
pcd_mainDataWordArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS     The information words were successfully determined.
LPS_FAILURE     The information words were not determined.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ---------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
NONE

NOTES:
None

End Unit pcd_MainDeterminePcdWord
*****************************************************************
```

```
UNIT NAME:
        pcd_MainDetermineScenes

PURPOSE:
        Identify the WRS scenes

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    The WRS scene(s) was successfully identified
LPS_FAILURE    The WRS scene(s) were not successfully identified
PCD_NODATA     Not enough data for interpolation

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE
Name                       Use Description
-------------------------- --- ---------------------------------------------

EXTERNAL VARIABLES: NONE
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: NONE

End Unit pcd_MainDetermineScenes
*****************************************************************
```

```
UNIT NAME:
         pcd_MainDetSceneDescription

PURPOSE:
        Determines the nominal scene center, the sun azimuth, sun elevation,
         and horizontal display shift

PARAMETERS:
pcd_mainPositionArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    The nominal scene center was identified and described
               successfully
LPS_FAILURE    The nominal scene ceneter was not identified and described
               successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES: NONE
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: 1. The calibration door status of the previous pcd node will be used as
          the calibration door status at the actual scene center.

End Unit pcd_MainDetSceneDescription
*****************************************************************
```

```
UNIT NAME:
        pcd_MainEphemerisInt

PURPOSE:
        Interpolate the ephemeris data point using the Lagrange interpolation
method

PARAMETERS:
pcd_mainEphTimeArg : data_in
pcd_mainEphDataArg : data_in
pcd_mainTimeArg : data_in
pcd_mainNumPtrArg : data_in
pcd_mainIntEphArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE
Name                       Use Description
-------------------------- --- -----------------------------------------------

EXTERNAL VARIABLES: NONE
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: NONE

End Unit pcd_MainEphemerisInt
****************************************************************
```

```
UNIT NAME: pcd_MainEstimateCycleTime

PURPOSE: To estimate a PCD cycle time

PARAMETERS:
pcd_mainPrevCycleArg : data_in
pcd_mainCurrCycleArg : data_inout

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
None

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) The previous PCD cycle time is valid.

NOTES:
None

End Unit pcd_MainEstimateCycleTime
*****************************************************************
```

```
UNIT NAME: pcd_MainEstimateMissingCycles

PURPOSE: To estimate the number of missing PCD cycles prior to the current PCD
         cycle.

PARAMETERS:
pcd_mainPrevCycleArg : data_in
pcd_mainCurrCycleArg : data_in
pcd_mainMissingCyclesArg : data_out

RETURN VALUE:  Type = void

Value          Description
-------------- -------------------------------------------------------------
None

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) Both PCD cycles have valid times, and the current PCD cycle time is greater
   than the previous PCD cycle time.

NOTES:

End Unit pcd_MainEstimateMissingCycles
****************************************************************
```

```
UNIT NAME:
        pcd_MainEtmToGci

PURPOSE:
        Compute the transformation matrix from ETM+ to GCI

PARAMETERS:
pcd_mainQutnArg : data_in
pcd_mainEtmToGciMatrixArg : data_out

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE
Name                       Use Description
-------------------------- --- -----------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
pcd_Global      pcd_validSceneParms pcdValidSceneParmStruct
                                                     I   IAS data items neeeded
                                                         for WRS scene
                                                         identifications

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: The math algorithm is documented in algorithm #2 section of WRS Scene
       Identification Algorithms dated July 31, 1995.

End Unit pcd_MainEtmToGci
*****************************************************************
```

```
UNIT NAME: pcd_MainEvalMnfQuality

PURPOSE: Generate PCD major frame quality and accounting statistics for the
         indicated PCD minor frame.

PARAMETERS:
pcd_mainMinorFrameArg : data_in
pcd_mainMajorFrameArg : data_inout

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
None

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) The pcd_mainMajorFrameArg was initialized to NULL at structure creation.

NOTES:
None

End Unit pcd_MainEvalMnfQuality:
*****************************************************************
```

UNIT NAME: pcd_MainExtractCycleInfo

PURPOSE: Extracts bands present bits from a PCD cycle.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	The bands present bytes were extracted successfully.
LPS_FAILURE	The bands present bytes were not extracted successfully.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
None		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
pcd_Global	pcd_eocFlag	lpsBooleanType	I	Flag that indicates the end of contact period
pcd_Global	pcd_cycleList	pcdCycleInfoStruct	I	Pointer to a list of PCD Cycles
pcd_Global	pcd_validParms	pcdValidParmStruct	I	The valid parameters used for PCD processing

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) If either of the bytes that contain the bands present bits contain fill, then no bands present will be placed into the database for this cycle

NOTES:
1) When bandInfo is full OR when an end of contact period is received the band information will be written to the database.
2) bandInfoCount is a static variable that is initialized to LPS_NULL when declared. It is used as an index for bandInfo.
3) bandInfo is a static local variable that is initialized to LPS_NULL when declared.

End Unit pcd_MainExtractCycleInfo
****

UNIT NAME: pcd_MainExtractSceneParms

PURPOSE: Extract the ephemeris, attitude, calibration door activity status and the spacecraft time from each PCD Cycle.

PARAMETERS:
RETURN_CODE : control_out
pcd_mainSceneParmArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	The scene parms were extracted successfully
LPS_FAILURE	The scene parms were not successfully extracted
PCD_NODATA	Not enough ephemeris/attitude data for interpolation

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
None		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
pcd_Global	pcd_cycleList	pcdCycleStructPtr	I	Pointer to a list of PCD cycles

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. If invalid/missing ephemeris/attitude data points are detected, the interpolation will be performed to fill up the missing points. In order to ensure interpolation accuracy, six points are accumulated for interpolation.
2. If there are not enough valid data points within two pcd cycles (75%) to perform ephemeris/attitude interpolation, a warning message will be logged to indicate no scenes will be determined during this period.

NOTES: NONE

End Unit pcd_MainExtractSceneParms
****

UNIT NAME: pcd_MainExtractWords

PURPOSE: Extracts the replicated PCD data words from PCD byte stream.

PARAMETERS:
pcd_mainDataWordArg : data_out
pcd_mainShmMemAddrArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	The replicated PCD data words were successfully extracted
LPS_FAILURE	The replicated PCD data words were not successfully extracted.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
NONE.		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
pcd_Global	pcd_subIntvChangedFlag	lpsBooleanType	O	Flag that indicates a subinterval change
pcd_Global	pcd_subIntvList	pcdSubIntvStruct	O	The subinterval info structure

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS
1) The MFPS will not send less than 650 * 4 PCD Bytes per shared memory block. (650 = the number of vcdus per ETM+ major Frame).

NOTES:
failedVotes, dataBreakAcct, and pcdWord are static variables which are initialized in the declaration section.

End Unit pcd_MainExtractWords
****

```
UNIT NAME:  pcd_MainFillMajorFrame

PURPOSE:    Insert fill for the indicated minor frames into the current PCD major
            frame node.

PARAMETERS:
pcd_mainMajorFrameArg : data_in
pcd_mainStartMnfNoArg : data_in
pcd_mainEndMnfNoArg : data_in

RETURN VALUE:  Type = void

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
pcd_Global      pcd_validParms       pcdValidParmStruct
                                                      I  A structure containing
                                                         the PCD processing
                                                         parameters

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
1) mnfFill is a static local buffer containing a fully filled minor frame

End Unit pcd_MainFillMajorFrame:
****************************************************************
```

```
UNIT NAME:  pcd_MainFillMultiCycles

PURPOSE:    Generate filled PCD cycle nodes for missing PCD cycles.

PARAMETERS:
RETURN_CODE : control_out
pcd_mainMissingCyclesArg : data_in

RETURN VALUE:  Type = void

Value          Description
-------------- ------------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
pcd_Global      pcd_cycleList        pcdCycleStructPtr
                                                     I/O Pointer to list of PCD
                                                         cycles

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:

End Unit pcd_MainFillMultiCycles:
*****************************************************************
```

```
UNIT NAME:  pcd_MainFillPcdCycle

PURPOSE:    Insert fill for the indicated major frames into the current PCD cyle
            node.

PARAMETERS:
pcd_mainEndMjfNoArg : data_in
pcd_mainStartMjfNoArg : data_in
pcd_mainPcdCycleArg : data_inout

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
None

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
pcd_Global      pcd_validParms       pcdValidParmStruct
                                                      I  A structure containing
                                                         validated PCD
                                                         parameters

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:

End Unit pcd_MainFillPcdCycle:
****************************************************************
```

```
UNIT NAME:  pcd_MainFormatPcdCycle

PURPOSE:    Extract and format the information available in the indicated major
            frame and place it into the current PCD cycle node.

PARAMETERS:
pcd_mainPcdCycleArg : data_inout
pcd_mainMajorFrameArg : data_in

RETURN VALUE:  Type = void

Value          Description
-------------- -------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ----------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: NONE

End Unit pcd_MainFormatPcdCycle:
****************************************************************
```

UNIT NAME: pcd_MainFormatWords

PURPOSE: Formats the PCD data words into data stream

PARAMETERS:
pcd_mainShmMemAddrArg : data_in
pcd_mainDataBreakAcctArg : data_inout
pcd_mainWordArg : data_inout

RETURN VALUE:
Type = void

Value	Description
None	

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
None		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
pcd_Global	pcd_validParms	pcdValidParmStruct	I	The PCD frame fill pattern
pcd_Global	pcd_validThres	pcdValidThresStruct	I	The PCD threholds

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
1) missingCount is a static local variable that retains the count of missing VCDUs (or PCD bytes). When the threshold is reached, a message is reported, and the counter is reset. A threshold is reported based on a modulo of the threshold.

End Unit pcd_MainFormatWords
****

```
UNIT NAME: pcd_MainInit

PURPOSE: Initializes the PCDS software for normal payload correction data
         processing.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS     The PCDS software has been successfully initalized.
LPS_FAILURE     The PCDS software has not been successfully initalized.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
pcd_Global      pcd_lpsConfig       lpsConfigStruct  I  LPS configuration
                                                          information

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:  NONE

NOTES:  NONE

End Unit pcd_MainInit
*****************************************************************
```

```
UNIT NAME:
        pcd_MainLocateSyncPatterns

PURPOSE:
        Locates the PCD word cycle sync patterns in the PCD word stream.

PARAMETERS:
pcd_mainWordArg : data_in
RETURN_CODE : control_out
pcd_mainSyncPosArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    All sync patterns were successfully located/identified.
LPS_FAILURE    All sync patterns wer not successfully located/identified.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:

End Unit pcd_MainLocateSyncPatterns
*****************************************************************
```

```
UNIT NAME:
        pcd_MainPerformMajorityVote

PURPOSE:
        Performs a majority vote on the replicated PCD Information Word.

PARAMETERS:
pcd_mainTripDataWordArg : data_in
pcd_mainMajVoteDataWordArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    A successful majority vote was performed
LPS_FAILURE    There was a majority vote failure (fill returned in argument)

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
PCD_Global      pcd_validParms       pcdValidParmStruct
                                                      I  Validated PCD parameter
                                                         structure

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
   A majority vote failure is considered to be one of the following
      - Three data words are avail, all 3 data words are different
      - Two data words are avail, both data words are different
      - Only one data word is available

   When each available data word is different, the first data word is
      the information word that will be used to build the PCD minor
      frame.

NOTES:
1) When PCDS moves to implementation, the majority voting scheme should be
   examined for improvement.  If the MFPS provides, along with the PCD bytes,
   the BCH data quality of the VCDU containing the PCD bytes, the accuracy of
   the majority voting could be considerably improved.  Let's take for example,
   the situation where we have two identical data words and one different data
   word.  If the BCH quality of the two identical data words were poor and the
   single data word was extracted from a clean VCDU, we should consider using
   the single data word.  Other scenarios exist as well.  It could be argued
   that this routine should be heavily dependent on the BCH flags.

End Unit pcd_MainPerformMajorityVote
*****************************************************************
```

UNIT NAME: pcd_MainReportScenes

PURPOSE: Reports the scene description to the database and to the IDPS.

PARAMETERS:
pcd_mainSceneDescrArg : data_in
RETURN_CODE : control_out

RETURN VALUE: Type = RETURN_CODE

Value	Description
LPS_SUCCESS	The scene description was reported successfully
LPS_FAILURE	The scene description was not reported successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
None		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
pcd_Global	pcd_eocFlag	lpsBooleanType	I	End of contact period indicator
pcd_Global	pcd_subIntvChangedFlag	lpsBooleanType	I	Indicates subinterval change

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) This routine is called at the end of a contact to report the final scene descriptions to the database.

NOTES:
1) sceneList is a static variable

End Unit pcd_MainReportScenes
****

```
UNIT NAME:
        pcd_MainSunAzimuthElev

PURPOSE:
        Compute the sun azimuth and elevation angles

PARAMETERS:
pcd_mainSunAzArg : data_out
pcd_mainSunElevArg : data_out
pcd_mainActSceneArg : data_in

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE
Name                       Use Description
-------------------------- --- -----------------------------------------------

EXTERNAL VARIABLES: NONE
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:
 1. The math algorithm is documented in Algorithm #3 in the Appendix C of LPS SRS
    dated April 28, 1995.

End Unit pcd_MainSunAzimuthElev
*****************************************************************
```

```
UNIT NAME:
        pcd_MainSunPos

PURPOSE:
        Determine the position of the sun in GCI coordinates

PARAMETERS:
pcd_mainSunPosArg : data_out
pcd_mainJulTimeArg : data_in

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE
Name                       Use Description
-------------------------- --- -----------------------------------------------

EXTERNAL VARIABLES: NONE
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:
 1. This unit is converted from the Fortran version of the FDF unit, i.e. SOL.
 2. The math algorithm is from the Explanatory Supplement to the Astronomical
Ephemeris and the American Ephemeris and Nautical Almanac (1961), page 98.

End Unit pcd_MainSunPos
*****************************************************************
```

```
UNIT NAME: pcd_MainUpdateSubIntv

PURPOSE: Updates the subinterval PCD file list with information for new
         subintervals.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:  Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS     The subinterval file was updated successfully
LPS_FAILURE     The subinterval file was not updated successfully

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -----------------------------------------------
PCD_File                   O  The PCD L0R file.

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ----------------------
pcd_Global      pcd_subIntvList     pcdSubIntvStruct I  PCD subinterval
                                                         information
pcd_Global      pcd_lpsConfig       lpsConfigStruct  I  LPS configuration
                                                         information

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) The MFPS will always commit the major frame stop time for a sub-interval
   prior to sending a new subinterval id and committing a new sub-interval
   start time.

NOTES:

End Unit pcd_MainUpdateSubIntv
****************************************************************
```

UNIT NAME: pcd_MainUpdateSubIntvStats

PURPOSE: Extract PCD accounting information from the current PCD cycle.

PARAMETERS:
pcd_mainSubIntvIdArg : data_in
RETURN_CODE : control_out
pcd_mainPcdCycleArg : data_in

RETURN VALUE:
Type = void

Value	Description
None	

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
None		

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
pcd_Global	pcd_subIntvList	pcdSubIntvStruct	I/O	A structure containing the PCD subintervl information

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) The pcd_subIntvList structure is initialized to NULL upon initialization of PCDS
2) If the subinterval ID is not found for current PCD cycle, then this unit reports a warning message and does not maintain the statistics for the current PCD cycle.
3) The number of major frames per subinterval include the fill major frames.

NOTES:
None

End Unit pcd_MainUpdateSubIntvStats
****

```
UNIT NAME: pcd_MainVerifyCycle

PURPOSE: Performs validation on the current PCD cycle

PARAMETERS:
pcd_mainPcdCycleArg : data_inout
pcd_mainMissingCyclesArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    The PCD cycle was verified successfully
LPS_FAILURE    The PCD cycle time could not be verified nor estimated

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None

NOTES:
1) prevCycle is a static local variable that contains the previous PCD cycle.
   It is initialized to LPS_NULL the first time into this unit.

End Unit pcd_MainVerifyCycle
*****************************************************************
```

```
UNIT NAME: pcd_MainVerifyCycleTime

PURPOSE: Verify the PCD spacecraft time and estimate time if time is invalid.

PARAMETERS:
pcd_mainPrevCycleArg : data_in
pcd_mainCurrCycleArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
PCD_TIMEVALID  The PCD cycle time was determined to be correct
PCD_TIMEINVALID
               The PCD cycle time was determined to be incorrect
PCD_TIMEJUMP   The PCD cycle time had a backward jump

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- ------------------- --------------- --- -----------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
None
NOTES:

End Unit pcd_MainVerifyCycleTime
****************************************************************
```

UNIT NAME: pcd_MainWriteCycleToFile

PURPOSE: Determines the correct sub-interval file for the current PCD Cycle

PARAMETERS:
pcd_mainSubIntvIdArg : data_out
RETURN_CODE : control_out
pcd_mainPcdCycleArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	The Cycle was successfully written to a file
LPS_FAILURE	The Cycle was not written to a file

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
PCD_File	O	The PCD L0R File
PCD_Trouble_File	O	The PCD trouble PCD Cycle File

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
pcd_Global	pcd_subIntvList	pcdSubIntvStruct	I	A structure that contains the PCD subinterval information

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1) A trouble file will be used to store the PCD Cycles that do not fit into any of the PCD sub-interval files.

NOTES:
1) fileSize is a static local variable initialized to 0 upon unit initialization. The value contains the PCD trouble file size in bytes

End Unit pcd_MainWriteCycleToFile:
****

```
UNIT NAME:  rdc_CalcAcctInfo

PURPOSE:    This unit calculates the accounting information for data transfer.

PARAMETERS:
rdc_calcTransTimeArg : data_out
rdc_calcTransRateArg : data_out
rdc_calcSizeOfBufferArg : data_in
rdc_calcNumberOfBuffersArg : data_in
rdc_calcActualStopTimeArg : data_in
rdc_calcActualStartTimeArg : data_in
RETURN_CODE : control_out
rdc_calcExpectedVolArg : data_out
rdc_calcReceivedVolArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

End Unit rdc_CalcAcctInfo
****************************************************************
```

```
UNIT NAME:  rdc_ConnectToDevice

PURPOSE:    This unit connects to the capture device and maps into Kernel
            memory.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    Success if connected to the capture device
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

End Unit rdc_ConnectToDevice
*****************************************************************
```

```
UNIT NAME:  rdc_db_WriteAcctToDb

PURPOSE:    This unit inserts the RDCS accounting information into the LPS
            database

PARAMETERS:
rdc_dbWriteAcctFDArg : data_in
rdc_dbWriteSchedIdArg : data_in
rdc_dbWriteCptrModeArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Accounting information is successfully written to the database.
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ---------------------------------------------
rdc_AcctFile                I  File containing RDCS accounting information
RDC_Acct                    O  Table in which RDC accounting information is
                               entered
LPS_Configuration           I  Table containing configuration items
Contact_Schedules           I  Scheduled start and stop times

EXTERNAL VARIABLES:
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit rdc_db_WriteAcctToDb
*****************************************************************
```

```
UNIT NAME:  rdc_db_WriteOnLineFlag

PURPOSE:    This unit either sets the OnLineFlag to LPS_TRUE or LPS_FALSE
            depending on the argument, rdc_dbOnLineArg

PARAMETERS:
rdc_dbOnLineFNArg : data_in
rdc_dbOnLineArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ----------------------------------------------------------------
LPS_SUCCESS    Successful setting of OnLineFlag in the database
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -------------------------------------------------
RDC_Acct                   O  Capture accounting table in the LPS database

EXTERNAL VARIABLES:
None.

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit rdc_db_WriteOnLineFlag
*****************************************************************
```

```
UNIT NAME:  rdc_DeIsolateProcess

PURPOSE:    This unit resets the transfer process to preemptive,
            sets this process to normal priority,
            and de-assigns the restricted processor.

PARAMETERS:

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
None

End Unit rdc_DeIsolateProcess
*****************************************************************
```

```
UNIT NAME:  rdc_DeleteRDCFiles

PURPOSE:    This program deletes the raw wideband data file and the
            associated accounting file.

PARAMETERS:
RETURN_CODE : control_out
rdc_deleteDataFileOnlyArg : data_in
rdc_deleteFileNameArg : data_in

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Success if raw data file is deleted
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ---------------------------------------------
rdc_AcctFile               I  The RDC accounting file
Raw_WB_Sets               N/A File with raw wideband data deleted

EXTERNAL VARIABLES:
None.

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:  The user is given a choice to delete the capture file
        whether the accounting information has been updated
        to the database or not.  If the files have not been
        updated to the database, the associated accounting file
        is not deleted.

End Unit rdc_DeleteRDCFiles
****************************************************************
```

```
UNIT NAME: rdc_GenDataRcvSumReport

PURPOSE:  Generate the LPS Data Receive Summary Report

PARAMETERS:
RETURN_CODE : control_out
rdc_rptContactSeqIdArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS
LPS_FAILURE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                     Use    Description
-----------------------  ---    ------------------------------------------------
-
LPS_Configuration         I     Database table containing LPS configuration
information
RDC_Acct                  I     Database table containing raw data capture
information
                                from the RDCS on a contact basis

EXTERNAL VARIABLES:
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
1. The Oracle Reports is used to generate the LPS Data Receive Summary report
   defined in this specification.
2. Connection/Disconnection to/from the database is through the Oracle Reports.

NOTES:

End Unit rdc_GenDataRcvSumReport PDL:
*****************************************************************
```

```
UNIT NAME:  rdc_GenLabel

PURPOSE:    This software generates the information necessary to print a label
            for removable media and invokes the software to print that label.

PARAMETERS:
rdc_labelDeviceArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    Successful printing of removable medium label
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
rdc_AcctFile               I  Capture accounting file

EXTERNAL VARIABLES:
None.

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
It is assumed that a removable medium is in the device.
This design is for single rdc_GenLabel processing.
There is only one capture accounting file on the removable
medium.

NOTES:

End Unit rdc_GenLabel
*****************************************************************
```

```
UNIT NAME:  rdc_Init

PURPOSE:    This unit initializes either the raw data capture to disk (rdc_Main)
            or the writing to the capture device  (rdc_TestData).

PARAMETERS:
rdc_initCptrModeArg : data_in
rdc_initFileNameArg : data_inout
rdc_initStopTimeArg : data_inout
RETURN_CODE : control_out
rdc_initAcctFNArg : data_out
rdc_initFileDescArg : data_out
rdc_initStringIdArg : data_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
rdc_PIDTempFile             O  This unit creates this temporary file and reads
                               the process Id into it.  This file is used to
                               determine that RDC is running.

Raw Wideband Sets          N/A Create raw data capture file for write

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

Capture/data write is allowed to continue with or without the following:

   - valid stop time
   - successful open of accounting file or temporary PID file
   - enough space on the capture disk for the longest contact duration.

NOTES:
        1.  If the stop time is not valid, defaults to maximum contact length.
        2.  String Id is acquired by the system command "gethostname."
        3.  An accounting file is created for the writing of test data as well
            as for the contact capture file.

End Unit rdc_Init
*****************************************************************
```

```
UNIT NAME:  rdc_IsolateProcess

PURPOSE:    This unit sets the transfer process to nonpreemptive,
            sets this process to nondegrading highest priority,
            and assigns this process to a restricted processor.

PARAMETERS:
rdc_processorNumArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:  The return_status is set to LPS_FAILURE only if setting the process to
        nonpreemptive, setting the process to nondegrading highest priority and
        setting the process to restricted processor fail.

End Unit rdc_IsolateProcess
*****************************************************************
```

```
UNIT NAME:  rdc_Main

PURPOSE:    This is the main program for the Raw Data Capture
            Subsystem. Raw data is captured to disk for later
            processing.

PARAMETERS:
rdc_mainSuspendGIDArg : data_in
rdc_mainSuspendArg : data_in
rdc_mainStopTimeArg : data_in
rdc_mainSchedIdArg : data_in
rdc_mainIsolateArg : data_in
rdc_mainFileNameArg : data_in
rdc_mainCptrModeArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Success in capturing raw data to disk
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
Raw data capture is allowed to continue with or without a number of requested
resources including access to the database.

NOTES:  Many of the units in this program are also invoked by rdc_TestData
        which sets up data to be captured for testing purposes.  When making
        changes to this software, care must be taken to monitor ramifications
        to rdc_TestData.

End Unit rdc_Main
*****************************************************************
```

```
UNIT NAME:  rdc_MainCapture

PURPOSE:    This unit accesses the device driver and
            writes the data to the capture disk.

PARAMETERS:
rdc_mainCptrModeArg : data_in
rdc_mainCptrFileDescArg : data_in
rdc_mainCptrDbFlagArg : data_in
rdc_mainCptrActualStopTimeArg : data_out
rdc_mainCptrActualStartTimeArg : data_out
rdc_mainCptrAcctFNArg : data_out
rdc_mainCptrExpectedVolArg : data_out
rdc_mainCptrStopTimeArg : data_in
rdc_mainCptrFileNameArg : data_inout
RETURN_CODE : control_out
rdc_mainCptrTransTimeArg : data_out
rdc_mainCptrTransRateArg : data_out
rdc_mainCptrReceivedVolArg : data_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Successful in capturing raw data
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ---------------------------------------------
Raw_WB_Sets                O  Disk file with raw data
rdc_PIDTempFile           N/A Temporary file used to indicate RDC activity.
                              This unit simply deletes this file.

EXTERNAL VARIABLES:
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

End Unit rdc_MainCapture
*****************************************************************
```

```
UNIT NAME:  rdc_PrintLabel

PURPOSE:    This unit generates a label for the removable medium.

PARAMETERS:
rdc_pLabelFNArg : data_in
rdc_pLabelChanIdArg : data_in
rdc_pLabelSchedIdArg : data_in
rdc_pLabelStartTimeArg : data_in
rdc_pLabelStopTimeArg : data_in
rdc_pLabelStringIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ---------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:  This unit's design is pending specification of tape label printer
        driver software.

End Unit rdc_PrintLabel
*****************************************************************
```

```
UNIT NAME:  rdc_ResetDevice

PURPOSE:    This unit resets the RDC_DEVICE.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Success if the RDC_DEVICE is reset
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:  Resetting the RDC_DEVICE  is  "black box" code written by the SGI
        engineer, Tom Potthast.
        There is a possible race condition with this activity.  This
        unit should be in a loop checking for RDC activity.  If RDC
        continues after this unit has been called, call rdc_ResetDevice
        again...

End Unit rdc_ResetDevice
*****************************************************************
```

```
UNIT NAME:  rdc_RestageCptr

PURPOSE:    This program uploads raw wideband data and capture accounting
            information from removable media for reprocessing.

PARAMETERS:
RETURN_CODE : control_out
rdc_restageDeviceArg : data_in

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Success if capture file is extracted from removable media
               and database is running.
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ----------------------------------------------
rdc_AcctFile               I  Capture accounting file
Raw_WB_Sets               N/A Capture file is extracted in this unit
rdc_PIDRestageFile         O  Contains restage process Id to facilitate
                              manual termination of this process.

EXTERNAL VARIABLES:

None.

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:  1.  This design handles only single raw data file restaging.
        2.  String Id is acquired by the system command "gethostname."
        3.  Once the accounting and raw data files are uploaded, manual
            termination is not allowed.  (PID file is deleted.)

End Unit rdc_RestageCptr
*****************************************************************
```

```
UNIT NAME:  rdc_ResumeProcess

PURPOSE:    This unit resumes the suspended Level 0R processes.

PARAMETERS:
rdc_suspendGIDArg : data_in

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

End Unit rdc_ResumeProcess
*****************************************************************
```

```
UNIT NAME:  rdc_Save

PURPOSE:    This program saves the captured raw wideband data to short-term
            storage along with the associated accounting information file.

PARAMETERS:
rdc_svDeviceArg : data_in
RETURN_CODE : control_out
rdc_svFileNameArg : data_in

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Raw_WB_Sets               N/A The captured raw wideband data file copied to
                              short_term storage removable media.
rdc_AcctFile              I/O Capture accounting information file
rdc_PIDSaveFile            O  File containing rdc_save Process Id used
                              used to facilitate manual termination of save

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:  This design handles only single raw data file saving.
        If this data has already been saved to removable media, this
        unit simply sends a message and continues to save.

End Unit rdc_save
*****************************************************************
```

```
UNIT NAME:  rdc_SetOnLineFlag

PURPOSE:    This program searches through the capture disk directory for
            raw wideband data files on-line and sets the on-line flag
            in the database.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Successful setting of on-line flag in the database
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Raw_WB_Sets               N/A Raw data capture file

EXTERNAL VARIABLES:
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
Restricted capture directory is known.

NOTES:

End Unit rdc_SetOnLineFlag
*****************************************************************
```

```
UNIT NAME:   rdc_ShutDownRDC

PURPOSE:     This program facilitates termination of RDC.

PARAMETERS:
rdc_shutDwnWaitOptArg : data_in
rdc_shutDwnFileNameArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =
Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Raw_WB_Sets                 I  Capture/Data write file
rdc_PIDTempFile             I  Temporary file signifying active RDC

EXTERNAL VARIABLES:
Source              Name            Type            Use Description
---------------     -------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

End Unit rdc_ShutDownRDC
*****************************************************************
```

```
UNIT NAME:  rdc_StopSaveRestage

PURPOSE:    This program sends a signal to the save or restage process to
            terminate the process.

PARAMETERS:
rdc_stpSvRsArg : control_in
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Termination of the requested process is successful.
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ---------------------------------------------
rdc_PIDSaveFile            I  File containing save process Id
rdc_PIDRestageFile         I  File containing restage process Id

EXTERNAL VARIABLES:
None.

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
This design is for single save/restage processing.
The files are in a restricted directory.

NOTES:

End Unit rdc_StopSaveRestage
*****************************************************************
```

```
UNIT NAME:  rdc_SuspendProcess

PURPOSE:    This unit suspends all Level 0R processes currently running.

PARAMETERS:
rdc_suspendGIDArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

End Unit rdc_SuspendProcess
*****************************************************************
```

UNIT NAME: rdc_TermRestageSig

PURPOSE: This unit is a signal handler. When invoked, it simply shuts down the restage process following disconnect from the database.

PARAMETERS:

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Successful deletion of restaged files and disconnection from the database
LPS_FAILURE	Failure to delete restaged files or to disconnect from the database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
None.

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
There is only one (or partial) raw data capture file and its associated accounting information file in the restricted directory for restaging.

NOTES:

End Unit rdc_TermRestageSig
****

```
UNIT NAME:  rdc_TermSaveSig

PURPOSE:    This unit is a signal handler.  When invoked, it simply shuts
            down the save process following disconnect from the database.

PARAMETERS:

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Successful disconnection from the database
LPS_FAILURE    Disconnect error

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ----------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:  The files (if any) are not deleted from the removable medium.

End Unit rdc_TermSaveSig
*************************************************************
```

```
UNIT NAME:  rdc_TestData

PURPOSE:    This is the main program for writing data to the capture device.
            Raw data is written to the capture device for testing raw data
            capture.

PARAMETERS:
rdc_testSuspendGIDArg : data_in
rdc_testSuspendArg : data_in
rdc_testStopTimeArg : data_in
rdc_testSchedIdArg : data_in
rdc_testIsolateArg : data_in
rdc_testFileNameArg : data_in
rdc_testCptrModeArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:  This program calls many of the same functions as rdc_Main
        which is the capture program. When making changes to this software, care
        must be taken to monitor ramifications to rdc_Main.

End Unit rdc_TestData
*****************************************************************
```

```
UNIT NAME:  rdc_testOutDataTrans

PURPOSE:    This unit accesses the device driver for data write.

PARAMETERS:
rdc_testCptrModeArg : data_in
rdc_testDTExpectedVolArg : data_out
rdc_testDTFileDescArg : data_in
RETURN_CODE : control_out
rdc_testDTTransTimeArg : data_out
rdc_testDTTransRateArg : data_out
rdc_testDTReceivedVolArg : data_out
rdc_testDTActualStopTimeArg : data_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
Raw_WB_Sets                 O  Raw wideband data file
rdc_PIDTempFile            N/A Temporary file used to indicate RDC activity.
                               This unit simply deletes this file.
EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit rdc_TestDataTrans
*****************************************************************
```

```
UNIT NAME:  rdc_UpdateDb

PURPOSE:    The accounting information from the data transfer is saved in a
            file to disk.  This program retrieves that information from the disk
            file and updates the LPS database.

PARAMETERS:
rdc_udbSchedIdArg : data_in
rdc_udbCptrModeArg : data_in
rdc_udbAcctFNArg : data_in
rdc_udbDbFlagArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Successfully updates the database with the capture accounting
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------------
rdc_AcctFile                O

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- -----------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

End Unit rdc_UpdateDb
****************************************************************
```

```
UNIT NAME:  rdc_UpdRDCAcct

PURPOSE:    This program searches for capture accounting information files
            in the reserved directory and updates the database.

PARAMETERS:
RETURN_CODE : control_out

RETURN VALUE:
Type =  RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Success if all of the files are updated to the database
LPS_FAILURE    Failure if any one of the files are not updated to the database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
rdc_AcctFile              I/O RDC accounting file

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit rdc_UpdRDCAcct
****************************************************************
```

```
UNIT NAME:  rdc_WritetoAcctFile

PURPOSE:    This unit simply creates the RDC accounting file and writes
            the rdc_Main/rdc_TestData accounting information to it.

PARAMETERS:
rdc_mainCptrExpectedVolArg : data_in
RETURN_CODE : control_out
rdc_writeTransTimeArg : data_in
rdc_writeTransRateArg : data_in
rdc_writeSuspendArg : data_in
rdc_writeStringIdArg : data_inout
rdc_writeStopTimeArg : data_in
rdc_writeIsolateArg : data_in
rdc_writeFileNameArg : data_in
rdc_writeReceivedVolArg : data_in
rdc_writeCptrModeArg : data_in
rdc_writeActualStopTimeArg : data_in
rdc_writeActualStartTimeArg : data_in
rdc_writeAcctFNArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
rdc_AcctFile                O  The RDC accounting file

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
None

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

End Unit rdc_WritetoAcctFile
*************************************************************
```

```
UNIT NAME: rdp_BCHBuildMsnQuadTable

PURPOSE: Reads in Mission Quad Table from rdp_BCHQuadFile.
         Table is used to solve Locator Polynomial.

PARAMETERS:
RETURN_CODE  :  control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Good return
LPS_FAILURE    Unable to read/open rdp_BCHQuadFile

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- -----------------------------------------------
rdp_BCHQuadFile            I  A 4 megabyte file that contains the pre-calculted
                              table.  It takes almost 9 minutes of CPU time to
                              create this file.

EXTERNAL VARIABLES:

Source          Name                Type                      Use Description
--------------- ------------------- ---------------           --- --------------
rdp_BCHTables   rdp_bchMsnQuadTable lpsUInt16Type             O  Table used to
                                    [rdp_BCHMSNWRDLEN - 1]       solve quad-
                                    [rdp_BCHMSNWRDLEN - 1]       radic eqns
                                    [2] (three dimensions)        for the Msn
                                                                  locator poly

NOTES:

End Unit rdp_BCHBuildMsnQuadTable
****************************************************************
```

```
UNIT NAME:  rdp_BCHCorrectBitsMsn

PURPOSE:Corrects Bits In Mission Codeblocks

PARAMETERS:
rdp_bchMsnCodeBlockArg : data_inout
rdp_bchMsnErrorArg : data_in
rdp_bchMsnItemCountArg : data_in

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
This is a static, local routine that will be packaged with
rdp_BCHMsnChienSearch

End Unit rdp_BCHCorrectBitsMsn
*****************************************************************
```

```
UNIT NAME:rdp_BCHCorrectBitsPtr

PURPOSE:Corrects Bits In Pointer Codeblock

PARAMETERS:
rdp_bchPtrBuffArg : data_inout
rdp_bchPtrErrorArg : data_in
rdp_bchPtrItemCountArg : data_in

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
This is a static, local routine that will be packaged with
rdp_BCHPtrChienSearch.

End Unit rdp_BCHCorrectBitsPtr
*****************************************************************
```

UNIT NAME:rdp_BCHDecode

PURPOSE:Main routine for BCH Error Detection & Correction. BCH error detection and Correction will be done only if CRC check NOT passed OR rdp_bchMustBCHArg is TRUE OR Reed Solomon Errors Found OR Synch Pattern errors found OR Bit slip encountered and the CADU is NOT a fill CADU. In the event that NO BCH error detection is employed, the BCH annotations for the CADU will be given LPS_BCH_NOT_PERFORMED statuses.

PARAMETERS:
rdp_bchMustBCHArg : control_in

rdp_bchThresholdArg : data_in
rdp_bchShmPtrArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Successful Operation
LPS_FAILURE	Failure because it could not open/read quad_file.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
Rdp_Stats	rdp_Stats	Structure	I/O	Maintains statistics relevant to this contact period.
rdp_MainFSyncGlobals	rdp_mfsSharedMem	lpsAddressType	I	Gives the address of the beginning of the shared memory block

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
BCH error detection & Correction will be done only if CRC check NOT passed OR rdp_bchMustBCHArg is TRUE OR Reed Solomon Errors Found OR Synch Pattern errors found OR Bit slip encountered and the CADU is NOT a fill CADU. In the event that NO BCH error detection is employed, the BCH annotations for the CADU will be given a LPS_BCH_NOT_PERFORMED statuses.

NOTES:
BCH error detection & Correction will be done only if CRC check NOT passed OR rdp_bchMustBCHArg is TRUE OR Reed Solomon Errors Found OR Synch Pattern errors found OR Bit slip encountered and the CADU is NOT a fill CADU. In the event that NO BCH error detection is employed, the BCH annotations for the CADU will be given a LPS_BCH_NOT_PERFORMED statuses.

End Unit rdp_BCHDecode
****

UNIT NAME:rdp_BCHMsnCalcSyndromes

PURPOSE:Calculate Syndromes for Mission Codeblocks

PARAMETERS:
RETURN_CODE : control_out
rdp_bchMsnCodeBlockArg : data_in

RETURN VALUE:
Type =unsigned long

Value	Description
Range of unsigned long	The three syndromes, each 10 bits long are concatenated. S1 occupies bits 0 - 9, S3 occupies bits 10-19 and S5 occupies bits 20-29. The last two bits are binary zero filled and are to be ignored.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
rdp_BCHTables	rdp_bchMsn_syn_table	lpsUInt32Type [128][256]	I	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).

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
* This function will read mission codewords and calculate syndromes S1,S3 &
* S5 by looking up codeword segments in the lookup table,
* rdp_bchMsn_syn_tabl.
* This is called by rdp_BCHTransposeCadu which deinterleaves the codewords.

End Unit rdp_BCHMsnCalcSyndromes
****

```
UNIT NAME: rdp_BCHMsnChienSearch

PURPOSE: Finds the roots to Error locator Polynomial for mission codeblocks

PARAMETERS:
rdp_bchSigma1Arg : data_in
rdp_bchSigma2Arg : data_in
rdp_bchELArg : data_in
rdp_bchSigma3Arg : data_in
rdp_bchMsnCodeBlockArg : data_inout
rdp_bchMsnErrorArg : data_out
rdp_bchMsnRootCountArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = lspUInt32Type
Value          Description
-------------- ------------------------------------------------------------------
lspUInt32Type  S1 occupies bits 0 - 9.  S3 occupies bits 10 - 19 and S5
               occupies bits 20 - 29.  The last two bits are binary zero
               filled and should be ignored.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
rdp_BCHTables   rdp_bchMsnQuadTable lpsUInt16Type    O      Table used to
                                    [rdp_BCHMSNWRDLEN - 1]  solve quadratic
                                    [rdp_BCHMSNWRDLEN - 1]  eqns for the Msn
                                    [2] (three dimensions)  locator poly

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
The routines, rdp_BCHReduceToQuadMsn and rdp_BCHMsnCorrectBits are local,
static routines used here only.

End Unit rdp_BCHMsnChienSearch
*************************************************************
```

```
UNIT NAME:rdp_BCHMsnDecTree

PURPOSE: Creates Mission locator polynomial for Chien Search
PARAMETERS:
rdp_bchSigma1Arg : data_out
rdp_bchMsnSyndromesArg : data_in
rdp_bchSigma2Arg : data_out
rdp_bchSigma3Arg : data_out
rdp_bchELArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE
Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    Good return
LPS_FAILURE    Error return

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
RDP_BCHTables   rdp_bchMsn_pow_tbl[2][RDP_BCHMSNWRDLEN]
                                     lpsUInt16Type    I  rdp_bchMsn_pow_tbl[0][x]
                                                        gives the 3rd power of
                                                        x in the Galois Field
                                                        GF(2^10 - 1)
                                                        rdp_bchMsn_pow_tbl[1][x]
                                                        gives the 5th power of
                                                        x in the Galois Field
RDP_BCHTables  rdp_bchMsn_gf[RDP_BCHMSNWRDLEN]       I  Powers of alpha.the
                                                        generator polynomial for
                                                        the Galois Field
                                                        GF(2^10 - 1)
RDP_BCHTables  rdp_bchMsn_gf_alt[RDP_BCHMSNWRDLEN]   I  Inverse table to
                                                        rdp_bchMsn_gf
End Unit rdp_BCHMsnDecTree
*************************************************************
```

```
UNIT NAME:rdp_BCHMsnDivide

PURPOSE:To divide one element of a Galois field by another.  (Mission Field).

PARAMETERS:
rdp_bchNumeratorArg : data_in
rdp_bchDenominatorArg : data_in
RETURN_CODE :  control_out

RETURN VALUE:
Type =lpsUInt16Type

Value          Description
-------------- -------------------------------------------------------------
0-1023         Value of result

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit rdp_BCHMsnDivide
****************************************************************
```

UNIT NAME:rdp_BCHPtrCalcSyndromes

PURPOSE:Calculate Syndromes for Pointer Codeblocks

PARAMETERS:
rdp_bchPtrBuffArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = unsigned short

Value	Description
Range of unsigned short	The three syndromes, each 5 bits long are concatenated. S1 occupies bits 0 - 4, S3 occupies bits 5-9 and S5 occupies bits 10-14. The last bit is a binary zero filled and is to be ignored.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
rdp_BCHTables	rdp_bchPtr_syn_table	lpsUInt16Type [4][256]	I	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).

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: None

NOTES:
* This function will read pointer codewords and calculate syndromes S1,S3 &
* S5 by looking up codeword segments in the lookup table,
* rdp_bchPtr_syn_table.
* This is called by rdp_BCHTransposeCadu which deinterleaves the codewords.

End Unit rdp_BCHPtrCalcSyndromes
****

```
UNIT NAME: rdp_BCHPtrChienSearch

PURPOSE: Finds the roots to Error locator Polynomial for Pointer codeblocks.

PARAMETERS:
rdp_bchSigma1Arg : data_in
rdp_bchSigma2Arg : data_in
rdp_bchSigma3Arg : data_in
rdp_bchPtrBuffArg : data_inout
rdp_bchPtrErrorArg : data_out
rdp_bchPtrRootCountArg : data_out
RETURN_CODE : control_out
rdp_bchELArg : data_in

RETURN VALUE:
Type = lpsUInt16Type
Value          Description
-------------- ------------------------------------------------------------------
lpsUInt16Type  S1 occupies bits 0 - 4.  S3 occupies bits 5 - 9 and S5
               occupies bits 10 - 14.  The last bit is a  binary zero
               filled and should be ignored.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ---------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
rdp_BCHTables   rdp_bchPtrQuadTable lpsUInt16Type    O           Table used to
                                     [RDP_BCHPTRWRDLEN - 1]      solve quad-
                                     [RDP_BCHPTRWRDLEN - 1]      radic eqns
                                     [2] (three dimensions)       for the Ptr
                                                                  locator poly
NOTES:
The routines, rdp_BCHReduceToQuadPtr and rdp_BCHPtrCorrectBits are local,
static routines used here only.

End Unit rdp_BCHPtrChienSearch
*****************************************************************
```

```
UNIT NAME:rdp_BCHPtrDecTree

PURPOSE: Creates Pointer locator polynomial for Chien Search
PARAMETERS:
rdp_bchPtrSyndromesArg : data_in
rdp_bchSigma1Arg : data_out
rdp_bchSigma2Arg : data_out
rdp_bchSigma3Arg : data_out
rdp_bchELArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE
Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Good return
LPS_FAILURE    Error return

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
RDP_BCHTables   rdp_bchPtr_pow_tbl[2][RDP_BCHPTRWRDLEN]
                                     lpsUInt16Type    I  rdp_bchPtr_pow_tbl[0][x]
                                                         gives the 3rd power of
                                                         x in the Galois Field
                                                         GF(2^5 - 1)
                                                         rdp_bchPtr_pow_tbl[1][x]
                                                         gives the 5th power of
                                                         x in the Galois Field

RDP_BCHTables  rdp_bchPtr_gf[RDP_BCHPTRWRDLEN]        I Powers of alpha. the
                                                         generator polynomial for
the
                                                         Galois Field GF(2^5 - 1)
RDP_BCHTables  rdp_bchPtr_gf_alt[RDP_BCHPTRWRDLEN]    I Inverse table to
                                                         rdp_bchPtr_gf
End Unit rdp_BCHPtrDecTree
*************************************************************
```

```
UNIT NAME:rdp_BCHPtrDivide

PURPOSE:To divide one element of a Galois field by another (Pointer Field).

PARAMETERS:
rdp_bchNumeratorArg : data_in
rdp_bchDenominatorArg : data_in
RETURN_CODE :  control_out

RETURN VALUE:
Type = lpsUInt16Type

Value          Description
-------------- -------------------------------------------------------------
0-31           Value of result
EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit rdp_BCHPtrDivide
****************************************************************
```

```
UNIT NAME:  rdp_BCHReduceToQuadMsn

PURPOSE:Solves Quadratic Equation (Mission Field) by Using a lookup Table

PARAMETERS:
rdp_bchMsnErrorArg : data_inout
rdp_bchSigma1Arg : data_in
rdp_bchSigma2Arg : data_in
rdp_bchMsnStartArg : data_in
rdp_bchMsnRootCntArg : data_inout

RETURN VALUE:
Type = void

Value          Description
-------------- -------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use      Description
--------------- -------------------- --------------- ---      --------------------
rdp_BCHTables   rdp_bchMsnQuadTable lpsUInt16Type    O       Table used to
                                     [rdp_BCHMSNWRDLEN - 1]       solve quad-
                                     [rdp_BCHMSNWRDLEN - 1]       radic eqns
                                     [2] (three dimensions)       for the Msn
                                                                  locator poly

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
This is a static, local routine that will be packaged with
rdp_BCHMsnChienSearch.

End Unit rdp_BCHReduceToQuadMsn
*************************************************************
```

```
UNIT NAME:rdp_BCHReduceToQuadPtr

PURPOSE:Solves Quadratic Equation (Pointer Field) by Using a lookup Table

PARAMETERS:
rdp_bchPtrErrorArg : data_inout
rdp_bchSigma1Arg : data_in
rdp_bchSigma2Arg : data_in
rdp_bchPtrStartArg : data_in
rdp_bchPtrRootCntArg : data_inout

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------------
NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------
rdp_BCHTables   rdp_bchPtrQuadTable lpsUInt16Type    O  Table used to
                                     [rdp_BCHPTRWRDLEN - 1]       solve quad-
                                     [rdp_BCHPTRWRDLEN - 1]       radic eqns
                                     [2] (three dimensions)       for the Ptr
                                                                  locator poly

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
This is a static, local routine that will be packaged with
rdp_BCHPtrChienSearch.

End Unit rdp_BCHReduceToQuadPtr
****************************************************************
```

```
UNIT NAME:rdp_BCHTransposeCadu

PURPOSE:Transpose CADU and Correct Codeblocks If Necessary.

PARAMETERS:
RETURN_CODE : control_out
rdp_bchCADUArg : data_inout

RETURN VALUE:
Type = void

Value          Description
-------------- ------------------------------------------------------------------
NONE

Name             Type                Use Description
---------------- ------------------- --- ---------------------------------------
rdp_bchCADUArg lpsAnnotatedCADUStruct *
                                     I/O Pointer to latest Annotated CADU in
                                         shared memory.  BCH annotations will
                                         be set in this routine

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                        Use Description
-------------------------- --- ------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
rdp_BCHTables   rdp_bchTrans_Table  lpsInt64Type[256]
                                                     I  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

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit rdp_BCHTransposeCadu
*****************************************************************
```

UNIT NAME:rdp_CRCCheck

PURPOSE: To check for bit errors based on the cyclic redundancy checksum code

PARAMETERS:
rdp_crcThresholdsArg : data_in
rdp_crcFirstEntryArg : control_in
rdp_crcShmPtrArg : data_in

RETURN VALUE:

Source	Name	Type	Use	Description
RDP_Stats	CADU_CRC_Err_Cnt	Int	O	Count of CADUs with CRC errors

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES: NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES: NONE

End Unit rdp_CRCCheck
****

```
UNIT NAME:rdp_CRCCheckSum

PURPOSE:Calculates the CRC checksum to to be used to check for bit errors in
 the raw wideband data.

PARAMETERS:
rdp_crcRegFlagArg : data_in
lpsUInt16Type : data_out
rdp_crcTableArg : data_in
rdp_crcVCDUArg : data_in
rdp_crcOctetsArg : data_in

RETURN VALUE:
Type = lpsUInt16Type

Value          Description
-------------- ------------------------------------------------------------------
(short int)    The computed 16 bit polynomial for the VCDU

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:NONE

EXTERNAL VARIABLES: NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: The CRC lookup table must have been
 generated by rdp_CRCGenTable prior to running this program.

NOTES: This unit is extracted from the generalized CRC checksum calculator,
 crc_16_double, written by Jacob Rosenberg, NASA/GSFC.

End Unit rdp_CRCCheckSum
*****************************************************************
```

```
UNIT NAME:rdp_CRCGenTable

PURPOSE:Builds the 16 bit lookup table based on a 17 bit polynomial to be used
 in the calculation of the CRC checksum

PARAMETERS:
rdp_crcPolyArg : data_in
rdp_crcTableArg : data_out

RETURN VALUE:NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:NONE

EXTERNAL VARIABLES: NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:NONE

NOTES: The code for this unit is reused in its entirety, except for name changes,
 from build_double_table written by Jacob Rosenberg, NASA/GSFC
The CRC generator polynomial is g(x)=x^16+x^12+x^5+1

End Unit rdp_CRCGenTable
*****************************************************************
```

```
UNIT NAME:
        rdp_db_GetCCSDSParms

PURPOSE:
        This routine retrieves the CCSDS Parameter table from the database
        server.

PARAMETERS:
RETURN_CODE : control_out
rdp_ccsdsParmArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Successfully retrieved the CCSDS_Parm table from the database
LPS_EDBERROR   Failed to retrieve a record due to database error
LPS_ENORECORD  No record found in the database

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ---------------------------------------------
Valid_CCSDS_Parms           I  CCSDS parameter table in database

EXTERNAL VARIABLES:
        NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
        1. The caller has connected to the database server prior to this call

NOTES:
        1. The caller is responsible for allocating a buffer to hold the
                returned Valid_CCSDS_Parms record

End Unit rdp_db_GetCCSDSParms
*************************************************************
```

```
UNIT NAME:rdp_db_GetThresholds

PURPOSE:Retrieve the thresholds for RDPS processing errors from the database

PARAMETERS:
RETURN_CODE : control_out
rdp_thresholdsArg : data_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Successful retrieval of RDP thresholds
LPS_FAILURE    Database error

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Valid_RDP_Thresholds       I  RDP error thresholds
lps_db_ErrorMessage        I  Get database error message

EXTERNAL VARIABLES: NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:NONE

NOTES:

End Unit rdp_db_GetThresholds
****************************************************************
```

```
UNIT NAME:rdp_db_PutRDPAcctInfo

PURPOSE:To store the return link accounting information in the database

PARAMETERS:
RETURN_CODE : control_out
rdp_mainCCSDSIndexArg : data_in
rdp_mainVersionArg : data_in
rdp_mainContactIdArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    No database errors
LPS_FAILURE    Database errors

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
RDP_Acct                    O  RDP accounting information database file
```

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
RDP_Stats	Inverted_CADU_Cnt	Int	I	Count of CADUs received with inverted polarity
	Polarity_Change_Cnt	Int	I	Count of CADU polarity changes
	CADU_Bit_Slip_Cnt	Int	I	Bit Slip Count
	CADU_Sync_Err_Cnt	Int	I	Sync Error Count
	CADU_Rcv_Cnt	Int	I	Received CADUs
	CADU_Flywheel_Cnt	Int	I	Flywheel Count
	Fill_CADU_Cnt	Int	I	Number of filled CADUs
	CADU_CRC_Err_Cnt	Int	I	CRC Error Count
	VCDU_Header1_Correctable_Err_Cnt	Int	I	Count of Corrected Headers,VCID1
	VCDU_Header2_Correctable_Err_Cnt	Int	I	Count of Corrected Headers,VCID2
	VCDU_Header_Uncorrectable_Err_Cnt	Int	I	Count of Uncorrected Headers
	BCH_Data_Corrected_Err_Cnt	Int	I	Count of BCH Corrected Data Fields
	BCH_Data_Uncorrected_Err_Cnt	Int	I	Count of BCH Uncorrected Data
	BCH_Data_Corrected_Bits_Cnt	Int	I	Count of bits corrected by BCH in data field
	BCH_Pointer_Corrected_Err_Cnt	Int	I	Count of BCH Corr. Pointer Fields
	BCH_Pointer_Uncorrected_Err_Cnt	Int	I	Count of BCH Uncorr. Pointers
	BCH_Pointer_Corrected_Bits_Cnt	Int	I	Count of bits corrected by BCH in pointer field

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES: NONE

End Unit rdp_db_PutRDPAcctInfo
****

UNIT NAME: rdp_Main

PURPOSE: Controls the identification, error detection and correction of CADUs for a given contact period

PARAMETERS:
rdp_mainDataFilenameArg : data_in
rdp_mainMustBCHArg : control_in
rdp_mainVersionArg : data_in
rdp_mainContactIdArg : data_in
RETURN_CODE : control_out

RETURN VALUE:None
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Successful raw data processing
LPS_FAILURE	Fatal error during RDPS processing

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
RDP_Globals	rdp_status	RETURN_CODE	O	Return status
RDP_Globals	rdp_failedFilePtr	lpsFilePtrType	I	Raw Data File Pointer

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: NONE

End Unit rdp_Main
****

```
UNIT NAME:rdp_MainExtractCADU

PURPOSE: Coordinate the task of obtaining raw data, locating cadus in raw
         data and putting them in a form that is suitable for futher
         processing.

PARAMETERS:
rdp_mainSyncThresholdArg : data_in
rdp_mainFirstEntryArg : control_in
rdp_mainCCSDSParmsArg : data_in
rdp_mainDataFilePtrArg : data_in
rdp_mainShmPtrArg : data_out
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure
RDP_EXCADUDONE Completion of all cadu extraction

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -----------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- ------------------------
lps_Globals     lps_shmRsrcKey      Natual           O  Key to be given
                                                        to lps_ShmGetWrBlk
                                                        to obtain write access
                                                        to shared memory
NOTES:
The vaiable, EOF_Flag, should be declared as a static because
its value must not change between invocations of this routine

End Unit rdp_MainExtractCADU
*****************************************************************
```

```
UNIT NAME:rdp_MainFSync

PURPOSE: Obtain Synch and Standardize All CADUs In a Physical Data Block

PARAMETERS:
rdp_mainSyncThresholdArg : data_in
rdp_mainRawBlockAddrArg : data_in
rdp_mainRawDataSizeArg : data_in
rdp_mainEOFArg : control_in
rdp_mainShmPtrArg : data_in
rdp_mainCCSDSParmsArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------------
LPS_SUCCESS    Entire Physical Block has been processed and shared memory has
               not overflowed; that is, a normal return.
LPS_FAILURE    Indicates an unrecoverable error.  Given when a failure value
               is returned from fs_frame_sync, fs_terminate or fs_initialize.
RDP_SHMBLKFULL Shared memory has overflowed.
RDP_FSYNCDONE  Frame synching is completely done.  That is, rdp_mainEOFArg is
               equal to LPS_TRUE and there is no overflow condition.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
NONE

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
rdp_MainFSyncGlobals rdp_mfsNumberHeldCadus lpsInt32Type
                                                    I/O Gives the number of
                                                        annotated CADUs held in
                                                        the array
                                                        rdp_mfsHoldCadus
rdp_MainFSyncGlobals rdp_mfsHoldCadus       lpsAddressType
                                                    I/O Holding buffer for
                                                        annotated CADUs that
                                                        could not be placed
                                                        into shared memory
rdp_MainFSyncGlobals rdp_SharedMemindex     lpsInt32Type
                                                    I/O Index into shared
                                                        memory.
rdp_MainFSyncGlobals rdp_mfsSharedMem       lpsAddressType
                                                    I/O Copy of
                                                        rdp_mainShmPtrArg to be
                                                        used by
                                                        rdp_MainTranToShared
rdp_MainFSyncGlobals rdp_mfsSyncThreshold   lpsInt32Type
                                                        Gives the number of
                                                        CADUs that have synch
                                                        errors that are
                                                        allowed before the
```

```
                                                          operator is informed
rdp_MainFSyncGlobals rdp_mfsBytesReceived  lpsInt64Type
                                                     I/O Number of bytes
                                                          in the physical blocks
                                                          PRIOR to the current
                                                          one.

End Unit rdp_MainFSync
*****************************************************************
```

```
UNIT NAME:rdp_MainGenerateOutput

PURPOSE: To generate RDPS output files

PARAMETERS:
rdp_mainFirstEntryArg : control_in
rdp_mainVersionArg : data_in
rdp_mainFailedFilePtrArg : data_in
rdp_mainCCSDSIndexArg : data_in
RETURN_CODE : control_out
rdp_mainContactIdArg : data_in
rdp_mainShmPtrArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Successful generation of output
LPS_FAILURE    Error in generation of output

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:
Source          Name                        Type            Use Description
--------------- -------------------         --------------- --- ----------------
RDP_Globals     rdp_shmKey                  lpsshmKeyType    I  Shared memory key

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: I

NOTES:

End Unit rdp_MainGenerateOutput
*****************************************************************
```

UNIT NAME:rdp_MainInit

PURPOSE: Initialize RDPS database, files and shared memory parameters.

PARAMETERS:
rdp_mainDataFilenameArg : data_in
rdp_mainCCSDSIndexArg : data_out
RETURN_CODE : control_out
rdp_mainCCSDSParmsArg : data_out
rdp_mainThresholdsArg : data_out
rdp_mainContactIdArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value	Description
LPS_SUCCESS	Successful completion
LPS_FAILURE	Fatal error

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
Raw_WB_Sets	N/A	Raw wideband data - opened for read
Failed_CADUs	N/A	Failed CADUs file - opened for write

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
RDP_Globals	rdp_dataFilePtr	Int	O	Raw data file pointer
RDP_Globals	rdp_failedFilePtr	Int	O	Failed file pointer
lps_Globals	LPS_SHMKEY_RDPMFP	Int	I	RDPS/MFPS shared memory key

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:NONE

End Unit rdp_MainInit
****

```
UNIT NAME: rdp_MainObtainData

PURPOSE:This unit retrieves a RDP_RAWBLOCKSIZE sized raw data block from a
        Raw_WB_Sets data file.

PARAMETERS:
rdp_mainRawDataSizeArg : data_out
rdp_mainRawBlockAddrArg : data_out
rdp_mainDataFilePtrArg : data_in
RETURN_CODE : control_out

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- -------------------------------------------------------------
LPS_SUCCESS    Success
LPS_FAILURE    Failure
RDP_RAWEOF     Indicate end-of-file

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                      Use Description
------------------------- --- ------------------------------------------------
Raw_WB_Sets                I  Raw Wide Band Data Set input file

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "rdp_MainObtainData"
****************************************************************
```

```
UNIT NAME:rdp_MainSetVCIDAnnotations

PURPOSE:Set annotations for fill CADUs and VCID change

PARAMETERS:
rdp_mainFirstEntryArg : control_in
rdp_mainShmPtrArg : data_in

RETURN VALUE: NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:NONE

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------
RDP_Stats       Fill_CADU_Cnt       Int              O  Number of Filled CADUs

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:NONE

NOTES:
 The first_CADU and previous_VCID flags are static variables.

End Unit rdp_MainSetVCIDAnnotations
*****************************************************************
```

UNIT NAME:rdp_MainShutdown

PURPOSE: Close down the database, files and shared memory areas used by RDPS

PARAMETERS:

RETURN VALUE: NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description
Raw_WB_Sets	I	Raw wideband sets file
Failed_CADUs	I	Failed CADUs trouble file

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
RDP_Globals	rdp_failedFilePtr	lpsFilePtrType	N/A	File descriptor for Failed CADUs file - Closed only
RDP_Globals	rdp_dataFilePtr	lpsFilePtrType	N/A	File descriptor for Raw WB Data file - Closed only
RDP_Globals	rdp_status	RETURN_CODE	N/A	Status of RDPS processing
lps_Globals	LPS_SHMKEY_RDPMFP	int	N/A	Shared memory key for RDPS/MFPS

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES: NONE

End Unit rdp_MainShutdown
****

```
UNIT NAME:rdp_MainStoreFailedCADUs

PURPOSE: To check CADUs stored in shared memory and write the failed
         (CRC/RS/BCH) CADUs to the open trouble file.

PARAMETERS:
rdp_mainFirstEntryArg : control_in
rdp_mainVersionArg : data_in
RETURN_CODE : control_out
rdp_mainFailedFilePtrArg : data_in
rdp_mainContactIdArg : data_in
rdp_mainShmPtrArg : data_in

RETURN VALUE:
Type = RETURN_CODE

Value          Description
-------------- ------------------------------------------------------------
LPS_SUCCESS    Successful execution
LPS_FAILURE    Error in writing to the disk

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- -------------------------------------------
Failed_CADUs                O  Failed CADUs file

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES:
      1. The storing is done on basis of one shared memory block.

End Unit rdp_MainStoreFailedCADUs
*****************************************************************
```

```
UNIT NAME: rdp_MainTranToShared

PURPOSE: This is the 'output' routine required by the frame synch software
         we are using.  It will transfer the annotated CADUs to the shared
         memory area.  In the event of an overflow, the CADUs will be stored
         into a holding area for subsequent transfer to the shared memory.
PARAMETERS:
FS_RETURN_CODE : control_out
rdp_mainAlignedCADUArg : data_in
rdp_mainNumFlippedBitsArg : data_in
rdp_mainInvertedCADUArg : data_in
rdp_mainSlippageArg : data_in
rdp_mainStateArg : data_in
rdp_mainPositionInBuffArg : data_in

RETURN VALUE:
Type = FS_RETURN_CODE
Value          Description
-------------- ------------------------------------------------------------------
LPS_FS_SUCCESS Good Return
LPS_FS_FAILURE Error encountered

Note that the identifiers 'LPS_SUCCESS' and 'LPS_FAILURE' cannot be used here
because this routine is called by a Rennaisance Building Block routine,  The
return codes must conform to what it is expecting.

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------
None
```

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
rdp_MainFSyncGlobals	rdp_mfsNumberHeldCadus	lpsInt32Type	I/O	Gives the number of annotated CADUs held in the array rdp_mfsHoldCadus
rdp_MainFSyncGlobals	rdp_mfsHoldCadus	lpsAddressType	I/O	Holding buffer for annotated CADUs that could not be placed into shared memory
rdp_MainFSyncGlobals	rdp_SharedMemindex	lpsInt32Type	I/O	Index into shared memory.
rdp_MainFSyncGlobals	rdp_mfsSharedMem	lpsAddressType	I/O	Copy of rdp_mainShmPtrArg to be used by rdp_MainTranToShared
rdp_MainFSyncGlobals	rdp_mfsSyncThreshold	lpsInt32Type		Gives the number of CADUs that have synch errors that are allowed before the operator is informed
rdp_MainFSyncGlobals	rdp_mfsBytesReceived	lpsInt64Type	I/O	Number of bytes in the physical blocks PRIOR to the current one.
rdp_Stats	rdp_Stats	Structure	I/O	Maintains statistics relevant to this contact period.

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:
It is assumed that the shared memory area is capable of holding more annotated CADUs than can be held in the raw input block plus the maximum number of flywheel frames permitted at a time.

NOTES: Pseudo-random noise is removed by the frame synch software itself. This routine has look-ahead logic, that is, it looks to see if space is available in the shared memory first. If space is available, it places the CADU in the shared memory area; otherwise, it places the CADU in the holding area. For clarity, the shared memory and the holding area are annotated CADU arrays.

End Unit rdp_MainTranToShared
****

```
UNIT NAME:rdp_MainValidateCADU

PURPOSE:Perform the grade 3 processing of the CADU

PARAMETERS:
rdp_mainThresholdsArg : data_in
rdp_mainFirstEntryArg : control_in
rdp_mainShmPtrArg : data_in

RETURN VALUE: NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES: NONE

EXTERNAL VARIABLES:NONE

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS: NONE

NOTES: NONE

End Unit rdp_MainValidateCADU
*****************************************************************
```

UNIT NAME:rdp_RSCheck

PURPOSE: To detect and correct errors based on the Reed-Solomon codewords

PARAMETERS:
rdp_rsThresholdsArg : data_in
rdp_rsFirstEntryArg : control_in
rdp_rsShmPtrArg : data_in

RETURN VALUE: NONE

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:

Name	Use	Description

EXTERNAL VARIABLES:

Source	Name	Type	Use	Description
RDP_Stats	VCDU_Header1_ Correctable_Err_Cnt	Int	O	Count of Correctable Headers for VCID1
	VCDU_Header2_ Correctable_Err_Cnt	Int	O	Count of Correctable Headers for VCID2
	VCDU_Header_ Uncorrectable_Err_Cnt	Int	O	Count of Uncorrectable Headers

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:
The arguments to rsd are set as follows for Landsat:
m=4
t=2
mo=6
poa=1
vf=5
i=1
rsmode=1

End Unit rdp_RSCheck
****

```
UNIT NAME:rsd

PURPOSE: Detect and correct errors in header using RS EDAC alogrithm

PARAMETERS:
rs_data : data_in

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
****************************************************************
```

```
UNIT NAME:RSDFree

PURPOSE: To free memory allocated for Reed-Solomon EDAC

PARAMETERS:

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                Type            Use Description
--------------- ------------------- --------------- --- -----------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
***************************************************************
```

```
UNIT NAME:RSDInit

PURPOSE:To initialize the Galois Field table and internal structure

PARAMETERS:
lpsInt32Type : data_out
gftPath : data_in
fsl : data_in
rsmode : data_in
I : data_in
vf : data_in
poa : data_in
mo : data_in
t : data_in
m : data_in

RETURN VALUE:
Type =

Value          Description
-------------- -------------------------------------------------------------

EXTERNAL FILE/RECORD/DBTABLE REFERENCES:
Name                       Use Description
-------------------------- --- ------------------------------------------------

EXTERNAL VARIABLES:
Source          Name                 Type            Use Description
--------------- -------------------- --------------- --- ------------------------

ASSUMPTIONS, CONSTRAINTS, RESTRICTIONS:

NOTES:

End Unit "Name"
****************************************************************
```