FMS E-Review 2005 Cover Sheet
09/04/14

INSPECTION COMPLETE

Load: A3240_REL2_TST_X02

Review ID:	58390.00	Life Cycle:	Low-Level Test Procedures	FAREA:	TESTS	DO-178 Level:	C
Review Type:	Standard Meeting	ACM Project:	FMS2000			Rework Effort (hours):	1.0
Produced:	HTS - Shanghai	ACM Subproject:	A3XX			Closure Effort (hours):	0.5

Review: Date | Time	Nov 4, 2014	10:00	Meeting Duration:	0.5
Review Location:	Moderator Cubicle		# Review Participants:	4
Conference Room:	N/A		Date Complete:	11/25/14
Telephone | Participant Code:	N/A	N/A	Review Status: (result of review)	Revise (No Further Review)

Moderator Closure → ✔
APPROVED
By Gu Ling at 10:00 am, Nov 25, 2014
Audit: Stamp Here

Some Work Deferred: ☐ Version(s) Unchanged: ☐

Work Product Type(s):

Component Test

Supporting Material(s) / Comments:

For: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE
11_10_10.SRD;12
11_10_4.SRD;31
11_10_1.SRD;47
11_21_3.SRD;62
11_2_1_7.SRD;42
PERF_TDRC_EXEC.SDD;106

Producer Employer:	Shanghai (FMS)	Oversight Eligible:	✔
# of Producer Technical Defects:	2	# of Oversight Focal Technical Defects:	0
# of Producer Non-technical Defects:	0	# of Oversight Focal Non-technical Defects:	0
# of Producer Process Defects:	0	# of Oversight Focal Process Defects:	0

Work Products Under Review

Reuse Scope: N/A

Problem Report	File Name	File Version	Review Size	Size Units	Approved Version
58390	CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF	2	5	test case	3
58390	CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.ZIP	3	9	undefined	4

Participants **Expert Pass-thru** ☐ **Comment**

Name	Function (discipline)/ Responsibility	Review Time (hours)	Role in review	Attend	Will Close	Signature check complete
Hu Zhimin	S/W Test	0.5	Producer	☒		✔ REVIEWED By Hu,Zhimin at 11:16 am, Nov 04, 2014
Gu, Ling	S/W Test	1.0	Moderator	☒	☒	✔ REVIEWED By Gu Ling at 3:14 pm, Nov 12, 2014
			Oversight	☐	☐	☐
Chen,Jixing	S/W Test	0.5	Reviewer	☒	☐	✔ REVIEWED By Chen,jixing at 11:02 am, Nov 20, 2014
	Process		Reviewer	☐	☐	☐

Assignee's signature (stamp) confirms that a review was performed and any action Items and markups were incorporated or dispositioned.
Participant's signature (stamp) confirms participation in the review. A lack of signature (stamp) indicates nonparticipation.
Moderator's signature (stamp) indicates record is complete.
Oversight Reviewer's signature (stamp) indicates that Supplier Oversight was conducted.
Safety Focal's signature (stamp) indicates partiicpation of a Safety Focal.

FIELD LEGEND:	Pre-Populated/Producer	Producer	Moderator	Participants	Oversight

Coversheet Continued

Name	Function (discipline)/ Responsibility	Review Time (hours)	Role in review	Attend	Will Close	Signature check complete
Qiong.Zhou	PDQE	1.0	Auditor	☒	☐	☑ REVIEWED By Zhou, Qiong at 9:57 am, Nov 05, 2014
Daryl.Duncan	PDQE		Auditor	☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐
				☐	☐	☐

FIELD LEGEND:	Pre-Populated/Producer	Producer	Moderator	Participants	Oversight

Component Test Procedure (CTP) Checklist **(CTP_CHECKLIST_WORD.doc 10/24/07)**	**ACM Project:**	**FMS2000**
	ACM Sub-Project:	**A3XX**
	SCR Number:	**58390.00**
	Affected Area:	**TESTS**

Overview: CTPs are generated to verify an individual software element or group of elements properly implement requirements the software element(s) trace to. Use this checklist to inspect test cases and associated test procedures, drivers, and stubs against requirements the software element(s) implement. The CTP(s) are verified to conform to standards, and fully test requirements with appropriate structural coverage. The associate tracing data and test coverage analysis/disposition data (if any) is also verified.

Misc Info Reference: FMS Test Process C71-5780-043, Section 5.

Yes	No	N/A	Administrative
◉	○	○	1. Do the CTPs elements follow the standard naming conventions? CTP_<A/C>_<FAREA>_<FUNC-NAME>.TDF file – CTP Test Definition File CTP_< A/C >_<FAREA>_< FUNC-NAME >.ZIP file – miscellaneous test related files CTP_< A/C >_<FAREA>_< FUNC-NAME>.TRT file – CTP Trace file(Core only)
			CTP elements configured in the CM tool:
◉	○	○	2. Is *.TDF file – CTP Test Definition File present?
◉	○	○	3. Is *.ZIP file present?
○	○	◉	4. Is *.TRT file – CTP Trace file present (Core only)?
			Review Packet information details:
◉	○	○	5. Is SCR Number and a copy of the SCR (Sec state) present?
◉	○	○	6. Is TDF, TRT(If present), ZIP files with correct generation information present?
◉	○	○	7. Support files (SRD, SDD, and Checklist) with Generation information.
◉	○	○	8. Does the review packet contain a difference listing of the old test to the new test and are the differences limited to the changes specified in this SCR?
◉	○	○	9. Is the version of the material under review and supporting material correct for the SCR(s)?
◉	○	○	10. Has the material/version been identified on the cover sheet of the review packet (may reference SCR)?
◉	○	○	11. Have all SCR fields (e.g. Analysis/Solution) been filled out properly?

Yes	No	N/A	TDF (CTP Test Definition File)

Does the TDF header include the following fields:

12. Does the TDF header include the following fields:
    - Filename
    - Title
    - Author
    - Creation Date
    - Modification History
    - Source
    - Description of TDF
13. Is the SCR number and description updated for this SCR?
14. Does the TDF header include a unique ANCHOR name for this CTP?
15. Is the list of SRD/SDD element references (and their generation numbers) updated and correct? (including formatting of this information)

Yes	No	N/A	ZIP File (CTP Related Miscellaneous Files)

16. Does the ZIP file contain the updated necessary test files ?
    - *.BAT
    - *.CUL
    - ~~*.DRV~~ (*_D.ADA)
    - ~~*.VER~~ (*.RST)
    - *.RPT
    - Optional files: STB, DSP, and INC (if necessary).
    - Has the *.CUL file been updated to show the correct span of source code procedures/functions that are being tested by this CTP?

Yes	No	N/A	TRT File (Core only)
◉	○	○	17. Does the TRT header include the following fields: • Filename • Title • Author • Creation Date • Modification History • Is the modification history with date, author, SCR number, and description updated?
◉	○	○	18. Has the traceability matrix been updated/verified (trace to the correct requirements)??

Yes	No	N/A	Test Case Design
◉	○	○	19. Are the test case ID numbers present in sequential order?
◉	○	○	20. Does the test script have test case descriptions which describe the objectives, intent, and operation for each test case?
◉	○	○	21. Are all the allocated requirements tested?
○	○	◉	22. If anchor is found to be a bad trace or vague/ambiguous, has it been disposed with a reference SCR.
◉	○	○	23. Does the test case description section of each test case identify the specific requirements (SRD anchors) that are being tested?
◉	○	○	24. Does the test case description section of each test case identify the specific requirements (SRD anchors) that are supporting requirements?
◉	○	○	25. To ensure robust testing, are all test cases inputs set with at least 2 different values?
◉	○	○	26. To ensure robust testing, are boundary conditions and tolerances tested where ever applicable?

Yes	No	N/A	Test Case Design con't
◉	○	○	27. Coverage Levels – Has every point of entry and exit in the program been invoked at least once?
◉	○	○	28. Coverage Levels – Has every decision in the program taken on all possible outcomes at least once?
○	○	◉	29. Coverage Levels – Has every condition in a decision in the program taken on all possible outcomes at least once?
○	○	◉	30. Coverage Levels – Has every condition in a decision been shown to independently affect that decision's outcome? A condition is shown to independently affect a decision's outcome by varying just that condition while holding fixed all other possible conditions.
◉	○	○	31. Data Coupling – Are there test cases which exercise "data coupling" between software modules (i.e., the dependence of a software component on data not exclusively under the control of that software component)?
◉	○	○	32. Data Coupling – Are there test cases which exercise "control coupling" between software modules (i.e., the manner or degree by which one software component influences the execution of another software component)?
○	○	◉	33. Error Guessing - Do areas in the software known to have complex algorithms have a sufficient number of test cases to ensure they are working as expected?
○	○	◉	34. Error Guessing - Do areas in the software associated with complex requirements have a sufficient number of test cases to ensure they are working as expected?
◉	○	○	35. Outputs - Are all test case outputs measured for at least two different values?
◉	○	○	36. Outputs - Have variables with expected output values been initialized to other values before input to the test process (e.g., If a variable is expected to have an output result of TRUE, is the input state of this variable set to FALSE before executing the test case?)
◉	○	○	37. Coverage Analysis - Are the entire test paths covered as per the structural coverage requirements mandated for Flight Management Systems? If not, are such structural coverage deficiencies dispositioned? If not determined to be a tool problem, then the disposition must reference to an SCR.
○	○	◉	38. Coverage Analysis - For uncovered requirements, is there another test that provides the coverage?
○	○	◉	39. Has the Test name and Anchor required if one exists, been identified? If not, has an SCR been written and the SCR number referenced?
○	○	◉	40. Coverage Analysis – Have all the failures been analyzed and disposed appropriately in the DSP quoting a correct SCR number documenting the reason for the failures.

Yes	No	N/A	Polymorphism Related Issues (C++)
○	○	◉	41. Has the code under test been examined for the existence of dynamic dispatch (can be determined by virtual functions in the code or a virtual table in the assembly code)?
○	○	◉	42. Does each test case appearing in the set of test cases associated with a class appear in the set of test cases associated with each of its subclasses?
○	○	◉	43. If dynamic dispatch is involved in the execution of a function, is the method separately tested in the context of every concrete class in which it appears, irrespective of whether it is defined by the class or inherited by it? An exception is made for simple get and set methods that only assign a value to, or return the value of an attribute or association. Such methods need only be tested once, in the context of the defining class.
◉	○	○	44. Are errors dispositioned to an SCR or has the test been updated?

Yes	No	N/A	Other
○	◉	○	45. Are all defects identified by the previous questions?

N N/A Justification Box

N/A:
For item 4, TRT file is not updated.
For item 22, there is no bad trace.
For item 29,30, this CTP test follows Level B standard.
For item 33,34, there are no "complex algorithms" or "complex requirements" for this CTP.
For item 38,39, there are no uncovered requirements in this CTP.
For item 40, there are no failures in this CTP.
For item 41-43, there is no C++ code or C++ stub code.
NO:
For item 45, please refer to check list item 45.

Change Category: PROBLEM SCR No.: P 58390.00
SCR Status: SEC SCR Status Date: 28-OCT-2014
Originator: Hu Zhimin Date Originated: 21-OCT-2014
Affected Area: TESTS Customer No.:
Assignee: Hu Zhimin Priority: 4
Verification Assignee: Gu, Ling
Found in Configuration: A340_REL2_SRC_CR2 Hardcopy Attachment: None
Target Configuration: A3240_REL2_TST_X02

Planned Impact: Test
Found During: SYS SPEC DEV/REVIEW
Aircraft Affected: A340
Task: N/A
CR1-F41 Type:

SCR Copied To: < None Entered >

SCR Copied From: < None Entered >

SCR Reissued To: < None Entered >

SCR Reissued From: < None Entered >

Title: Create new SCR

Description:

updated the CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF not for the requirement changed.

SRB Reviewed By: O'Connor, Michael Date: 24-OCT-2014

Analysis/Solution:

Updated CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE(TDF:2,ZIP:3)for A340 S2 CR2
SER430
on Build ST2099.
TDF(Gen=2)
1.Updated the TDF from the matrix to sequence format.
2.Updated SRD/SDD generation:
11_10_1.SRD;32-->47
11_2_1_7.SRD;31-->42

PERF_TDPC_EXEC.SDD;64-->106
3.Updated the PDB file from "D:\Database\PS203C-00077_88780000.o"
to "A:\Loadable_DBs\PDB_PS203C-00283_88780000.COFF".
ZIP(Gen=3)
1.Updated the Rst,Rpt,Dsp file.

Elements Affected:

Doc.	Element	Generation
TEST	CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF	2
TEST	CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.ZIP	3

ASSIGNEE: Hu Zhimin Date: 29-OCT-2014
VERIFIER: Date:
CCB COORDINATOR: Date:

< Close Cat/Dup SCR field continued > SCR No. 58390.00 Page 2 of 2

Closure Category: Fixed/Added Duplicate SCR No.: 00000.00
Project Status: Done
Addendum:
Visual Review Info:
Cert Concern:
Cust Notification:
Inservice Incident:
FDE Distraction:
Pilot Input:
Workload Wrkaround:
Must Fix:
Score/Comment:
Cause: N/A
Closed in Config.: A3240_REL2_TST_X02

Mode: All Lines

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF

```
1   1
2   2  FILE                    :  CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF
3   3
4   4  SOURCE CONFIGURATION    :  ISS (Instruction Set Simulator)
5   5
6   6  DESCRIPTION             :  This test is to verify that the variables are properly initialized.
7   7
8   8  MODIFICATION HISTORY    :
9   9                             DATE           SCR #          AUTHOR              DESCRIPTION
10  10                            =====          =====          ======              ===========
11  11
12  12                         Aug 12, 2010      52527.07       Yanfei Shen         Initial Development for A340 S1A S1 plan.
13  13                                                                              1. Rollover from A320
14  14                                                                                 CTP_A320_PERF_DESPATH_CALC_CC_RATE(TDF;7,
    » ZIP;5).
15  15                                                                              2. Updated the PDB file from A320
16  16                                                                                 "D:\Database\PDB_PS200H-08247_81476000.cof
    » f" to
17  17                                                                                 A340 "D:\Database\PS203C-00077_88780000.o"
    » .
18  18                                                                              4. Modified some expected value since the dat
    » abase changed.
    19
    20                          Oct 22, 2014      58390.00       Hu Zhimin           Updated for A340 S2 CR2 SER430 on Build ST209
    » 9.
    21                                                                              1.Updated the TDF from the matrix to sequence
    »  format.
    22                                                                              2.Updated SRD/SDD generation:
    23                                                                                 11_10_1.SRD;32-->47
    24                                                                                 11_2_1_7.SRD;31-->42
    25                                                                                 PERF_TDPC_EXEC.SDD;64-->106
    26                                                                              3.Updated the PDB file from "D:\Database\PS20
    » 3C-00077_88780000.o"
    27                                                                                 to "A:\Loadable_DBs\PDB_PS203C-00283_887800
    » 00.COFF".
19  28
20  29
21  30 SRD/SDD DETAILS:         11_10_10.SRD;12
22  31                          11_10_4.SRD;31
23      ~~                      11_10_1.SRD;32~~
    32                          11_10_1.SRD;47
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF (continued)

```
24  33                          11_21_3.SRD;62
25                              11_2_1_7.SRD;31
26                              PERF_TDPC_EXEC.SDD;64
    34                          11_2_1_7.SRD;42
    35                          PERF_TDPC_EXEC.SDD;106
27  36
28  37
29  38  TRACE DETAILS         :
30  39          ANCHOR        :  A340_PERF_TEST_2351
31  40
32  41          SOURCE        :  SRD; PERF_SRD_9751, PERF_SRD_9752, PERF_SRD_2721, PERF_SRD_2043, PERF_SRD_7473, PERF_SRD_9646
33  42                           SDD; PERF_SDD_1576, PERF_SDD_1577, PERF_SDD_1578, PERF_SDD_1579_INT,
34  43                                PERF_SDD_1580, PERF_SDD_1581_INT, PERF_SDD_3571
35  44
36  45  -- Note: Aero_Engine used in this CTP is A340-643 TRENT560 A340 model is used (Key ID :=2370)
37  46  ********************************************************************************************************
38  47  INITIALIZATIONS:
39  48
40  49
41  50  --FP_DEF_TOL = 0.001
42  51  FP_DEF_TOL = 0.001
43  52
44  53  --
45  54  define symbol True                                  := Standard.True
46  55
47  56  SUT_VARS
48  57  -- enumeration types
49  58  True
50  59
51  60  -- variables
52  61  Ctp_Perf_Despath_Calc_CC_Rate.Crzalt
53  62  Ctp_Perf_Despath_Calc_CC_Rate.Et
54  63  Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq
55  64  Ctp_Perf_Despath_Calc_CC_Rate.Critcralt
56  65
57  66  Perf_Background_Dpkg.Psdestalt
58  67  Perf_Background_Dpkg.Psdestisadev
59  68  Perf_Background_Dpkg.Pstropoalt
60  69  Perf_Background_Dpkg.Pcsttary.Stt.Isadev
61  70  Perf_Background_Dpkg.Pscabinrate
62  71  Perf_Background_Dpkg.Psdestqnh.Data
63  72  Ctp_Perf_Despath_Calc_CC_Rate.Rvs
64  73
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF (continued)

```
Pdb_Constants.Hreps01
Pdb_Constants.Hreps02
Pdb_Constants.Hreps03

Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_is_valid
Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_Data

END_SUT_VARS

DEFAULTS
END_DEFAULTS

-- This macro is the way to set up the PDB with the appropriate KEY no.

MACRO Set_Input
#sba Perf_Pdb_Initialization_Pkg.Init_Pdb after_elab
#go
Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_is_valid :=true
Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_Data :=%1

ENDMACRO

-- NOTES:
```

```
**********************************************************************
                                                    T E S T C A S E S
======================================================================================================================
» ==========
                                                                                                          1          2
» ----------  3
TEST COMMANDS                                                                                  NOTE   12345 67890 12345 67890
» 12345 67890
======================================================================================================================
» ==========
Test Case Description                                                                                 |      |ABCDE|.....|.....|.....|
» .....|.....|
----------------------------------------------------------------------------------------------------|------|-----|-----|-----|-----|
» -----|-----|
#download "D:\Database\PS203C-00077_88780000.o"                                                      | 1    |ABCDE|.....|.....|.....|
» .....|.....|
!Set_Input(2370)                                                                                    |      |ABCDE|.....|.....|.....|
» .....|.....|
Ctp_Perf_Despath_Calc_CC_Rate.Crzalt                    := 20000.0                                  |      |..CDE|.....|.....|.....|
» .....|.....|
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF (continued)

```
99                                                          := 22050.0          |      |.B...|.....|.....|.....|
  » .....|.....|
100                                                         := 22000.0          |      |A....|.....|.....|.....|
  » .....|.....|
101 Ctp_Perf_Despath_Calc_CC_Rate.Rvs                        := 0.0              |      |A..DE|.....|.....|.....|
  » .....|.....|
102 Ctp_Perf_Despath_Calc_CC_Rate.Et                         := 1.0              |      |A....|.....|.....|.....|
  » .....|.....|
103                                                         := 35.0             |      |.B...|.....|.....|.....|
  » .....|.....|
104                                                         := 20.000           |      |..C..|.....|.....|.....|
  » .....|.....|
105                                                         := 0.05             |      |...DE|.....|.....|.....|
  » .....|.....|
106 Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq                  := 0.0              |      |ABCDE|.....|.....|.....|
  » .....|.....|
107 Ctp_Perf_Despath_Calc_CC_Rate.Criteralt                  := 0.0              |      |ABCDE|.....|.....|.....|
  » .....|.....|
108 Perf_Background_Dpkg.Psdestalt                          := 200.0            |      |ABCDE|.....|.....|.....|
  » .....|.....|
109 Perf_Background_Dpkg.Psdestisadev                       := 2.0              |      |ABC..|.....|.....|.....|
  » .....|.....|
110                                                         := 2.1              |      |...DE|.....|.....|.....|
  » .....|.....|
111 Perf_Background_Dpkg.Pstropoalt                         := 36089.0          |      |ABCDE|.....|.....|.....|
  » .....|.....|
112 Perf_Background_Dpkg.Pesttary.Stt.Isadev                := 2.0              |      |ABCDE|.....|.....|.....|
  » .....|.....|
113 Perf_Background_Dpkg.Pscabinrate                        := 5.0              |      |..C..|.....|.....|.....|
  » .....|.....|
114                                                         := 40.0             |      |A....|.....|.....|.....|
  » .....|.....|
115                                                         := 29.0             |      |.B...|.....|.....|.....|
  » .....|.....|
116                                                         := 400.0            |      |...D.|.....|.....|.....|
  » .....|.....|
117                                                         := 600.0            |      |....E|.....|.....|.....|
  » .....|.....|
118 Perf_Background_Dpkg.Psdestqnh.Data                     := 1013.0           |      |.B...|.....|.....|.....|
  » .....|.....|
119                                                         := 1000.0           |      |A....|.....|.....|.....|
  » .....|.....|
120                                                         := 1100.0           |      |...DE|.....|.....|.....|
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF (continued)

```
          » .....|.....|
121                                                     := 450.0                |       |..C..|.....|.....|.....|
          » .....|.....|
122       # sba prf_despath_pkg.calc_cc_rate #127                               |       |A..DE|.....|.....|.....|
          » .....|.....|
123       # go                                                                  |       |A..DE|.....|.....|.....|
          » .....|.....|
124       # ctp_perf_Despath_calc_cc_rate.Rvs := rvs                            |       |A..DE|.....|.....|.....|
          » .....|.....|
125       !run_test()                                                           |       |ABCDE|.....|.....|.....|
          » .....|.....|
      97  *************************************************************************************************************************
          » ********
126   98
127       -- Outputs
128
129       Pdb_Constants.Hreps01                             = 0.872                 |       |ABCDE|.....|.....|.....|
          » .....|.....|
130       Pdb_Constants.Hreps02                             = 597.0                 |       |ABCDE|.....|.....|.....|
          » .....|.....|
131       Pdb_Constants.Hreps03                             = 7.000                 |       |ABCDE|.....|.....|.....|
          » .....|.....|
132       Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq           = 15.545                |       |A....|.....|.....|.....|
          » .....|.....|
133                                                         = 0.0                   |       |.B...|.....|.....|.....|
          » .....|.....|
134                                                         = 7.039                 |       |...D.|.....|.....|.....|
          » .....|.....|
135                                                         = 0.0                   |       |..C..|.....|.....|.....|
          » .....|.....|
136       Ctp_Perf_Despath_Calc_CC_Rate.Criteralt           = 3.34580E+03           |       |A....|.....|.....|.....|
          » .....|.....|
137                                                         = 2.20600E+04           |       |.B...|.....|.....|.....|
          » .....|.....|
138                                                         = 20010.0               |       |..C..|.....|.....|.....|
          » .....|.....|
139                                                         = 8.73624E+03           |       |...D.|.....|.....|.....|
          » .....|.....|
140       Ctp_Perf_Despath_Calc_CC_Rate.Rvs                 = 1200.0                |       |A....|.....|.....|.....|
          » .....|.....|
141                                                         = 1600.0                |       |...D.|.....|.....|.....|
          » .....|.....|
142                                                         = 2000.0                |       |....E|.....|.....|.....|
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF (continued)

```
                » ~~.....|.....|~~
143             ~~==========================================================================================~~
                » ~~==========~~
144             ~~==========================================================================================~~
                » ~~==========~~
145             ~~Test Case Description:~~
        99      TESTID: 1
146     100
147             ~~A.  Cc_Rate is calculated and repressurization is needed.  Destpress, Deltapress, and Stdcrzpress are all calculated c~~
                » ~~orrectly~~
        101         Cc_Rate is calculated and repressurization is needed.  Destpress, Deltapress, and Stdcrzpress are all calculated c
                » orrectly
148     102         but not able to be looked at due to debugger optimizations. Also Perf_sdd_3571 along with corresponding srd anchor
                » s are
149     103         tested implicitly. Are confirmed from end results.  RVS is set to min of 1200 ft/min.
150     104         PERF_SDD_1576(PERF_SRD_2043, PERF_SRD_9751, PERF_SRD_2721)
151     105         PERF_SDD_1577(PERF_SRD_9751)
152     106         PERF_SDD_1578(PERF_SRD_9751, PERF_SRD_9752)
153     107         PERF_SDD_1580(PERF_SRD_9751)
154     108         PERF_SDD_3571(PERF_SRD_7473, PERF_SRD_9646)
155     109
156             ~~B.  Et is such that Repressurization segment is too short.~~
        110
        111
        112     --    Load database.
        113     #download "A:\Loadable_DBs\PDB_PS203C-00283_88780000.COFF"
        114     !Set_Input(2370)
        115     Ctp_Perf_Despath_Calc_CC_Rate.Crzalt := 22000.0
        116     Ctp_Perf_Despath_Calc_CC_Rate.Rvs := 0.0
        117     Ctp_Perf_Despath_Calc_CC_Rate.Et := 1.0
        118     Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq := 0.0
        119     Ctp_Perf_Despath_Calc_CC_Rate.Critcralt := 0.0
        120     Perf_Background_Dpkg.Psdestalt := 200.0
        121     Perf_Background_Dpkg.Psdestisadev := 2.0
        122     Perf_Background_Dpkg.Pstropoalt := 36089.0
        123     Perf_Background_Dpkg.Pcsttary.Stt.Isadev := 2.0
        124     Perf_Background_Dpkg.Pscabinrate := 40.0
        125     Perf_Background_Dpkg.Psdestqnh.Data := 1000.0
        126     # sba prf_despath_pkg.calc_cc_rate #127
        127     # go
        128     # ctp_perf_Despath_calc_cc_rate.Rvs := rvs
        129     !run_test()
        130
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF (continued)

```
                131 -- Outputs
                132
                133
                134 Pdb_Constants.Hreps01 = 0.872
                135 Pdb_Constants.Hreps02 = 597.0
                136 Pdb_Constants.Hreps03 = 7.000
                137 Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq = 15.545
                138 Ctp_Perf_Despath_Calc_CC_Rate.Critcralt = 3.34580E+03
                139 Ctp_Perf_Despath_Calc_CC_Rate.Rvs = 1200.0
                140
                141
                142 TESTID: 2
                143
                144     Et is such that Repressurization segment is too short.
157             145     PERF_SDD_1576(PERF_SRD_2043, PERF_SRD_9751, PERF_SRD_2721)
158             146     PERF_SDD_1577(PERF_SRD_9751)
159             147     PERF_SDD_1578(PERF_SRD_9751, PERF_SRD_9752)
160             148     PERF_SDD_1580(PERF_SRD_9751)
161             149     PERF_SDD_1581_INT
162             150     PERF_SDD_3571(PERF_SRD_7473, PERF_SRD_9646)
163             151
164                     ~~C.  Et is such that Rptimereq is equal to 0.~~
                152
                153
                154 --    Load database.
                155 #download "A:\Loadable_DBs\PDB_PS203C-00283_88780000.COFF"
                156 !Set_Input(2370)
                157 Ctp_Perf_Despath_Calc_CC_Rate.Crzalt := 22050.0
                158 Ctp_Perf_Despath_Calc_CC_Rate.Et := 35.0
                159 Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq := 0.0
                160 Ctp_Perf_Despath_Calc_CC_Rate.Critcralt := 0.0
                161 Perf_Background_Dpkg.Psdestalt := 200.0
                162 Perf_Background_Dpkg.Psdestisadev := 2.0
                163 Perf_Background_Dpkg.Pstropoalt := 36089.0
                164 Perf_Background_Dpkg.Pcsttary.Stt.Isadev := 2.0
                165 Perf_Background_Dpkg.Pscabinrate := 29.0
                166 Perf_Background_Dpkg.Psdestqnh.Data := 1013.0
                167 !run_test()
                168
                169 -- Outputs
                170
                171
                172 Pdb_Constants.Hreps01 = 0.872
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF (continued)

```
Pdb_Constants.Hreps02 = 597.0
Pdb_Constants.Hreps03 = 7.000
Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq = 0.0
Ctp_Perf_Despath_Calc_CC_Rate.Critcralt = 2.20600E+04


TESTID: 3

    Et is such that Rptimereq is equal to 0.
    PERF_SDD_1576(PERF_SRD_2043, PERF_SRD_9751, PERF_SRD_2721)
    PERF_SDD_1577(PERF_SRD_9751)
    PERF_SDD_1578(PERF_SRD_9751, PERF_SRD_9752)
    PERF_SDD_1579_INT
    PERF_SDD_3571(PERF_SRD_7473, PERF_SRD_9646)

D.  Cc_Rate is calculated and repressurization is needed.  Destpress, Deltapress, and Stdcrzpress are all calculated c
» orrectly


--    Load database.
#download "A:\Loadable_DBs\PDB_PS203C-00283_88780000.COFF"
!Set_Input(2370)
Ctp_Perf_Despath_Calc_CC_Rate.Crzalt := 20000.0
Ctp_Perf_Despath_Calc_CC_Rate.Et := 20.000
Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq := 0.0
Ctp_Perf_Despath_Calc_CC_Rate.Critcralt := 0.0
Perf_Background_Dpkg.Psdestalt := 200.0
Perf_Background_Dpkg.Psdestisadev := 2.0
Perf_Background_Dpkg.Pstropoalt := 36089.0
Perf_Background_Dpkg.Pcsttary.Stt.Isadev := 2.0
Perf_Background_Dpkg.Pscabinrate := 5.0
Perf_Background_Dpkg.Psdestqnh.Data := 450.0
!run_test()

-- Outputs


Pdb_Constants.Hreps01 = 0.872
Pdb_Constants.Hreps02 = 597.0
Pdb_Constants.Hreps03 = 7.000
Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq = 0.0
Ctp_Perf_Despath_Calc_CC_Rate.Critcralt = 20010.0

```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF (continued)

	214	
	215	TESTID: 4
	216	
	217	Cc_Rate is calculated and repressurization is needed. Destpress, Deltapress, and Stdcrzpress are all calculated c
		» orrectly
172	218	but not able to be looked at due to debugger optimizations. Are confirmed from end results. Same as A but differ
		» ent values
173	219	to confirm calculations.
174	220	PERF_SDD_1576(PERF_SRD_2043, PERF_SRD_9751, PERF_SRD_2721)
175	221	PERF_SDD_1577(PERF_SRD_9751)
176	222	PERF_SDD_1578(PERF_SRD_9751, PERF_SRD_9752)
177	223	PERF_SDD_1580(PERF_SRD_9751)
178	224	PERF_SDD_3571(PERF_SRD_7473, PERF_SRD_9646)
179	225	
180		~~E. RVS is limited to 2000 ft/min.~~
	226	
	227	
	228	-- Load database.
	229	#download "A:\Loadable_DBs\PDB_PS203C-00283_88780000.COFF"
	230	!Set_Input(2370)
	231	Ctp_Perf_Despath_Calc_CC_Rate.Crzalt := 20000.0
	232	Ctp_Perf_Despath_Calc_CC_Rate.Rvs := 0.0
	233	Ctp_Perf_Despath_Calc_CC_Rate.Et := 0.05
	234	Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq := 0.0
	235	Ctp_Perf_Despath_Calc_CC_Rate.Critcralt := 0.0
	236	Perf_Background_Dpkg.Psdestalt := 200.0
	237	Perf_Background_Dpkg.Psdestisadev := 2.1
	238	Perf_Background_Dpkg.Pstropoalt := 36089.0
	239	Perf_Background_Dpkg.Pcsttary.Stt.Isadev := 2.0
	240	Perf_Background_Dpkg.Pscabinrate := 400.0
	241	Perf_Background_Dpkg.Psdestqnh.Data := 1100.0
	242	# sba prf_despath_pkg.calc_cc_rate #127
	243	# go
	244	# ctp_perf_Despath_calc_cc_rate.Rvs := rvs
	245	!run_test()
	246	
	247	-- Outputs
	248	
	249	
	250	Pdb_Constants.Hreps01 = 0.872
	251	Pdb_Constants.Hreps02 = 597.0
	252	Pdb_Constants.Hreps03 = 7.000
	253	Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq = 7.039

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF (continued)

	254	Ctp_Perf_Despath_Calc_CC_Rate.Critcralt = 8.73624E+03
	255	Ctp_Perf_Despath_Calc_CC_Rate.Rvs = 1600.0
	256	
	257	
	258	TESTID: 5
	259	
	260	RVS is limited to 2000 ft/min.
181	261	PERF_SRD_9751
182		~~==========~~
183		~~Note:~~
	262	
184	263	
185		~~1. Load database.~~
186		~~**********~~
	264	-- Load database.
	265	#download "A:\Loadable_DBs\PDB_PS203C-00283_88780000.COFF"
	266	!Set_Input(2370)
	267	Ctp_Perf_Despath_Calc_CC_Rate.Crzalt := 20000.0
	268	Ctp_Perf_Despath_Calc_CC_Rate.Rvs := 0.0
	269	Ctp_Perf_Despath_Calc_CC_Rate.Et := 0.05
	270	Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq := 0.0
	271	Ctp_Perf_Despath_Calc_CC_Rate.Critcralt := 0.0
	272	Perf_Background_Dpkg.Psdestalt := 200.0
	273	Perf_Background_Dpkg.Psdestisadev := 2.1
	274	Perf_Background_Dpkg.Pstropoalt := 36089.0
	275	Perf_Background_Dpkg.Pcsttary.Stt.Isadev := 2.0
	276	Perf_Background_Dpkg.Pscabinrate := 600.0
	277	Perf_Background_Dpkg.Psdestqnh.Data := 1100.0
	278	# sba prf_despath_pkg.calc_cc_rate #127
	279	# go
	280	# ctp_perf_Despath_calc_cc_rate.Rvs := rvs
	281	!run_test()
	282	
	283	-- Outputs
	284	
	285	
	286	Pdb_Constants.Hreps01 = 0.872
	287	Pdb_Constants.Hreps02 = 597.0
	288	Pdb_Constants.Hreps03 = 7.000
	289	Ctp_Perf_Despath_Calc_CC_Rate.Rvs = 2000.0

Mode: All Lines

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt

```
1  1  ##########################################################
2  2  #                                                        #
3  3  #                 Test Coverage Analyzer                 #
4  4  #            Short Summary Coverage Report               #
5  5  #                                                        #
6  6  ##########################################################
7  7
8     ~~                    Thu Sep 02 15:12:17 China Standard Time 2010~~
   8                        Wed Oct 22 13:28:17 China Standard Time 2014
9  9
10    ~~    AMD 29050 Test Coverage Analyzer (TCA)  V5.7.06 CLASS A ps4082880-107~~
11    ~~    Win32 Host: WinNT 5.1 Build 2600  UserID: E323675  Node: CH71DT3BKN32X (Intel PentPro Model 15 Step 6)~~
12    ~~    Current Dir: D:\A340_063\CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE~~
   10    Test Coverage Analyzer (TCA)  V6.15 CLASS A ps4082880-123
   11    Win32 Host: WinNT 6.1 Build 7601  UserID: e872753  Node: CH71DT25J7P02 (Intel PentPro Model 58 Step 9)
   12    Current Dir: C:\TEST_A340\CTP\update\CR2\CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE\new
13 13
14 14 ------------------------------------------------------------------------------
15    ~~TCA invoked Thu Sep 02 15:12:12 China Standard Time 2010 with command line:~~
16    ~~   tca.exe -TABS -r CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt -s -type 3 -p ...~~
   15 TCA invoked Wed Oct 22 13:28:12 China Standard Time 2014 with command line:
   16    tca.exe -TABS -r CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt -type 4 -p ...
17 17    CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_d.pth -x ...
18 18    CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.xin -c ...
19 19    CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.cul
20 20 ------------------------------------------------------------------------------
21 21 Expanded command line:
22    ~~   tca.exe -TABS -r CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt -s -type 3 -p ...~~
   22    tca.exe -TABS -r CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt -type 4 -p ...
23 23    CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_d.pth -x ...
24 24    CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.xin -c ...
25 25    CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.cul
26 26 ------------------------------------------------------------------------------
27 27
28 28
29    ~~Test Coverage Type: 3~~
   29 Test Coverage Type: 4
30 30
31 31 Report File Name  : CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt
32 32
33 33 Paths file(s) :
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt (continued)

```
34 34
35      (P01) CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_d.pth   Wed Jul 21 17:34:11 2010
   35   (P01) CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_d.pth   Wed Oct 22 13:27:16 2014
36 36              HADS-290x0 (PC/Windows NT) Ada Compiler, Version 2.9, PS4078711-104
37      ----------HADS-290x0 (PC/Windows NT) Ada Compiler, Version 2.9.56, PS4082845-106
38      ----------Post Object Paths Processor (POPP), v1.4, ps4082858-105
   37              HADS-290x0 (PC/Windows NT) Ada Compiler, Version 2.9.61, PS4082845-107
   38              Post Object Paths Processor (POPP), v1.6, ps4082858-107
39 39              Honeywell 29K Assembler , v3.9, ps4082836-115
40 40              Honeywell 29K Assembler, V2.4, PS4072677-105
41 41              Post Object Paths Processor (POPP), v1.3, ps4082858-104
42      ----------HADS-290x0 (PC/Windows NT) Ada Linker, Version 2.9.57, PS4082846-107
   42              Post Object Paths Processor (POPP), v1.4, ps4082858-105
   43              HADS-290x0 (PC/Windows NT) Ada Linker, Version 2.9.61, PS4082846-109
43 44
44 45 XInfo file(s)   Test Date   Test Platform:
45 46
46 47   (P01) CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_d.pth
47      --------(X01) CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.xin   Thu Sep 02 15:11:28 2010   ISS TCA Xinfo, Platform V7.02.04
   48           (X01) CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.xin   Wed Oct 22 13:27:56 2014   ISS TCA Xinfo, Platform V7.02.04
48 49
49 50 ------------------------------------------------------------------------------
50 51        Compilation                    Test Coverage Statistics       Warnings
51 52         Unit Name               Total Decision Cond Statemnt Block  Mixed Bool
52 53 ------------------------------ ------------------------------------ ----------
53      PRF_DESPATH_PKG.CALC_CC_RATE     100.0   100.0   n/a   100.0  100.0     0    0
54      ----------------------------------------4/4     n/a   17/17  18/18
   54 PRF_DESPATH_PKG.CALC_CC_RATE     100.0    n/a    n/a   100.0  100.0     0    0
   55                                           n/a    n/a   17/17  18/18
55 56
56 57 ------------------------------------------------------------------------------
57      Total Percentages                        100.0   n/a   100.0  100.0
58      Totals                                   4/4     n/a   17/17  18/18
   58 Total Percentages                         n/a    n/a   100.0  100.0
   59 Totals                                    n/a    n/a   17/17  18/18
59 60 Total Coverage                 100.0
60 61 ------------------------------------------------------------------------------
61 62 □
62 63 ********************************************************************************
63 64
64      --------AMD 29050 Test Coverage Analyzer (TCA)  Version 5.7.06 CLASS A
   65           Test Coverage Analyzer (TCA)  Version 6.15 CLASS A
65 66
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt (continued)

```
66  67 ********************************************************************************
67  68
68      ~~Coverage Type: 3~~
    69 Coverage Type: 4
69  70
70  71 Date of report / Report name :
71  72
72      ~~        Thu Sep 02 15:12:17 2010 CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt~~
    73         Wed Oct 22 13:28:17 2014 CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt
73  74
74  75 Current Directory:
75  76
76      ~~        D:\A340_063\CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE~~
    77         C:\TEST_A340\CTP\update\CR2\CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE\new
77  78
78  79 Paths file(s) :
79  80
80      ~~  (P01) CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_d.pth   Wed Jul 21 17:34:11 2010~~
    81   (P01) CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_d.pth   Wed Oct 22 13:27:16 2014
81  82             HADS-290x0 (PC/Windows NT) Ada Compiler, Version 2.9, PS4078711-104
82      ~~             HADS-290x0 (PC/Windows NT) Ada Compiler, Version 2.9.56, PS4082845-106~~
83      ~~             Post Object Paths Processor (POPP), v1.4, ps4082858-105~~
    83             HADS-290x0 (PC/Windows NT) Ada Compiler, Version 2.9.61, PS4082845-107
    84             Post Object Paths Processor (POPP), v1.6, ps4082858-107
84  85             Honeywell 29K Assembler , v3.9, ps4082836-115
85  86             Honeywell 29K Assembler, V2.4, PS4072677-105
86  87             Post Object Paths Processor (POPP), v1.3, ps4082858-104
87      ~~             HADS-290x0 (PC/Windows NT) Ada Linker, Version 2.9.57, PS4082846-107~~
    88             Post Object Paths Processor (POPP), v1.4, ps4082858-105
    89             HADS-290x0 (PC/Windows NT) Ada Linker, Version 2.9.61, PS4082846-109
88  90
89  91 XInfo file(s)   Test Date   Test Platform:
90  92
91  93   (P01) CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_d.pth
92      ~~          (X01) CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.xin   Thu Sep 02 15:11:28 2010   ISS TCA Xinfo, Platform V7.02.0~~
        » ~~4~~
    94          (X01) CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.xin   Wed Oct 22 13:27:56 2014   ISS TCA Xinfo, Platform V7.02.0
        » 4
93  95
94  96 Source file(s) :
95  97
96      ~~  V:\Integ_Builds\a340\S1A063\SRC_S1A063\FM\PRF_DESPATH_PKG_CALC_CC_RA -~~
97      ~~     TE.ADA~~
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt (continued)

	98	A:\a340_Builds\st2099\SRC_st2099\FM\PRF_DESPATH_PKG_CALC_CC_RATE.ADA
98	99	
99	100	Total Coverage statistics :
100	101	
101		~~TYPE 3, 100.0%~~
	102	TYPE 4, 100.0%
102	103	
103	104	
104	105	**********
105	106	Source Report Legend Key
106	107	(Legend Key may be suppressed by -k option)
107	108	
108	109	Coverage messages preceding source code lines are annotated with
109	110	object code block tags of the form [x-y BLOCKTYPE]. For example,
110	111	[263-17 JMPT] is a block tag for the 17th block of the 263rd unit
111	112	in the pathsfile and is a jump true block.
112	113	This block tag annotation is intended to be used as a reference to
113	114	the object code level block report (.tcb) generated with the -B option.
114	115	Each object code block is labeled with a unique block tag.
115	116	
116	117	Each line of source code may be prefixed by one of the following
117	118	indicators:
118	119	. = source line completely or partially executed
119	120	* = source line shown ONLY to clarify previous source lines and
120	121	is NOT actually part of the uncovered source TCA is reporting on
121	122	Note that no prefix indicates source line was not executed
122	123	
123	124	
124	125	**********
125	126	
126	127	Compilation Unit / Source file :
127	128	
128	129	PRF_DESPATH_PKG.CALC_CC_RATE
129		~~V:\Integ_Builds\a340\S1A063\SRC_S1A063\FM\PRF_DESPATH_PKG_CALC_CC_RA -~~
130		~~TE.ADA~~
	130	C:\A340\Builds\ST2099\SRC_ST2099\fm\PRF_DESPATH_PKG_CALC_CC_RATE.ADA
131	131	
132	132	Coverage statistics :
133	133	
134		~~TYPE 3, 100.0%~~
	134	TYPE 4, 100.0%
135	135	
136	136	Executed Total

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rpt (continued)

137		~~Decision Paths 4 4~~
138		~~Condition Paths n/a n/a~~
139	137	Statements 17 17
140	138	Blocks 18 18
141	139	
142	140	
143	141	
144	142	********** End of Report **********

Mode: All Lines

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rst

```
                                RESULTS FILE

********************************************************************************
                            Test Results Summary

          Percentage of Comparisons Passed    : 100.0000%

          Total Number of Comparisons Failed  : 0
          Total Number of Unknown Comparisons : 0
          Total Number of Comparisons Passed  : 26
          Total Number of Comparisons         : 26
          Total Number of Test Cases Included : 5

          Test Complete



********************************************************************************

~~Test Start Time: Sep 02 15:12:00 2010~~
Test Start Time: Oct 22 13:27:59 2014

FILE                  :  CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TDF

SOURCE CONFIGURATION  :  ISS (Instruction Set Simulator)

DESCRIPTION           :  This test is to verify that the variables are properly initialized.

MODIFICATION HISTORY  :
                         DATE           SCR #          AUTHOR              DESCRIPTION
                         =====          =====          ======              ===========

                      Aug 12, 2010      52527.07       Yanfei Shen        Initial Development for A340 S1A S1 plan.
                                                                          1. Rollover from A320
                                                                             CTP_A320_PERF_DESPATH_CALC_CC_RATE(TDF;7,
» ZIP;5).
                                                                          2. Updated the PDB file from A320
                                                                             "D:\Database\PDB_PS200H-08247_81476000.cof
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rst (continued)

```
                » f" to
40  40                                                                  A340 "D:\Database\PS203C-00077_88780000.o"
                » .
41  41                                                                4. Modified some expected value since the dat
                » abase changed.
42  42
    43                    Oct 22, 2014       58390.00       Hu Zhimin          Updated for A340 S2 CR2 SER430 on Build ST209
                » 9.
    44                                                                  1.Updated the TDF from the matrix to sequence
                »  format.
    45                                                                  2.Updated SRD/SDD generation:
    46                                                                     11_10_1.SRD;32-->47
    47                                                                     11_2_1_7.SRD;31-->42
    48                                                                     PERF_TDPC_EXEC.SDD;64-->106
    49                                                                  3.Updated the PDB file from "D:\Database\PS20
                » 3C-00077_88780000.o"
    50                                                                    to "A:\Loadable_DBs\PDB_PS203C-00283_887800
                » 00.COFF".
43  51
    52
44  53  SRD/SDD DETAILS:          11_10_10.SRD;12
45  54                            11_10_4.SRD;31
46      ~~                        11_10_1.SRD;32~~
    55                            11_10_1.SRD;47
47  56                            11_21_3.SRD;62
48      ~~                        11_2_1_7.SRD;31~~
49      ~~                        PERF_TDPC_EXEC.SDD;64~~
    57                            11_2_1_7.SRD;42
    58                            PERF_TDPC_EXEC.SDD;106
50  59
51  60
52  61  TRACE DETAILS          :
53  62          ANCHOR         :  A340_PERF_TEST_2351
54  63
55  64          SOURCE         :  SRD; PERF_SRD_9751, PERF_SRD_9752, PERF_SRD_2721, PERF_SRD_2043, PERF_SRD_7473, PERF_SRD_9646
56  65                            SDD; PERF_SDD_1576, PERF_SDD_1577, PERF_SDD_1578, PERF_SDD_1579_INT,
57  66                                 PERF_SDD_1580, PERF_SDD_1581_INT, PERF_SDD_3571
58  67
59  68  -- -- -- Note: Aero_Engine used in this CTP is A340-643 TRENT560 A340 model is used (Key ID :=2370)
60  69  -- -- --         BEGIN PROCESSING INCLUDE FILE C:\Program Files\honeywell_eng\TGS_v4_5_2\bin\debug_cmds.inc
61  70  -- -- --         END PROCESSING INCLUDE FILE C:\Program Files\honeywell_eng\TGS_v4_5_2\bin\debug_cmds.inc
62  71  -- -- -- ************************************************************
63  72  -- -- --                      INITIALIZATION SECTION
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rst (continued)

```
64  73 -- -- -- ************************************************************************
65  74 -- -- --FP_DEF_TOL = 0.001
66  75
67  76
68  77 CONSTANT                                                                                                  VALUE
69  78 ------------------------------------------------------------------------------------------------------  ------------------
       » --------
70  79 FP_DEF_TOL
       »    0.001
71  80
72  81
73  82 define symbol True                                  := Standard.True
74  83
75  84
76  85 CONSTANT                                                                                                  VALUE
77  86 ------------------------------------------------------------------------------------------------------  ------------------
       » --------
78  87 DBG_TIMEOUT
       »      300
79  88
80  89
81  90 TESTID: 1
82  91
83  92     Cc_Rate is calculated and repressurization is needed.  Destpress, Deltapress, and Stdcrzpress are all calculated c
       » orrectly
84  93     but not able to be looked at due to debugger optimizations. Also Perf_sdd_3571 along with corresponding srd anchor
       » s are
85  94     tested implicitly. Are confirmed from end results.  RVS is set to min of 1200 ft/min.
86  95     PERF_SDD_1576(PERF_SRD_2043, PERF_SRD_9751, PERF_SRD_2721)
87  96     PERF_SDD_1577(PERF_SRD_9751)
88  97     PERF_SDD_1578(PERF_SRD_9751, PERF_SRD_9752)
89  98     PERF_SDD_1580(PERF_SRD_9751)
90  99     PERF_SDD_3571(PERF_SRD_7473, PERF_SRD_9646)
91 100
92 101
93 102 INPUT                                                                                                   VALUE
94 103 ------------------------------------------------------------------------------------------------------  ------------------
       » --------
95 104 Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_is_valid
       »     true
96 105 Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_Data
       »     2370
97 106 Ctp_Perf_Despath_Calc_CC_Rate.Crzalt
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rst (continued)

```
      |       |» 22000.0
   98 |   107 |Ctp_Perf_Despath_Calc_CC_Rate.Rvs
      |       |»     0.0
   99 |   108 |Ctp_Perf_Despath_Calc_CC_Rate.Et
      |       |»     1.0
  100 |   109 |Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq
      |       |»     0.0
  101 |   110 |Ctp_Perf_Despath_Calc_CC_Rate.Critcralt
      |       |»     0.0
  102 |   111 |Perf_Background_Dpkg.Psdestalt
      |       |»   200.0
  103 |   112 |Perf_Background_Dpkg.Psdestisadev
      |       |»     2.0
  104 |   113 |Perf_Background_Dpkg.Pstropoalt
      |       |» 36089.0
  105 |   114 |Perf_Background_Dpkg.Pcsttary.Stt.Isadev
      |       |»     2.0
  106 |   115 |Perf_Background_Dpkg.Pscabinrate
      |       |»    40.0
  107 |   116 |Perf_Background_Dpkg.Psdestqnh.Data
      |       |»  1000.0
  108 |   117 |ctp_perf_Despath_calc_cc_rate.Rvs
      |       |»     rvs
  109 |   118 |
  110 |   119 |
  111 |   120 |OUTPUT                                                EXPECTED              TOLERANCE              ACTUAL
      |       |»          P/F
  112 |   121 |------------------------------------------------  ------------------------------  -------------  ------------------
      |       |» --------  ----
  113 |   122 |Pdb_Constants.Hreps01                                                    0.872          0.001                  8.7
      |       |» 2000E-01  P
  114 |   123 |Pdb_Constants.Hreps02                                                    597.0          0.001                  5.9
      |       |» 7000E+02  P
  115 |   124 |Pdb_Constants.Hreps03                                                    7.000          0.001                  7.0
      |       |» 0000E+00  P
  116 |   125 |Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq                                  15.545          0.001                  1.5
      |       |» 5452E+01  P
  117 |   126 |Ctp_Perf_Despath_Calc_CC_Rate.Critcralt                               3.34580E+03          0.001                  3.3
      |       |» 4580E+03  P
  118 |   127 |Ctp_Perf_Despath_Calc_CC_Rate.Rvs                                        1200.0          0.001                  1.2
      |       |» 0000E+03  P
  119 |   128 |
  120 |   129 |
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rst (continued)

```
121 130 ====> All 6 Comparisons Passed <====
122 131
123 132
124 133 TESTID: 2
125 134
126 135     Et is such that Repressurization segment is too short.
127 136     PERF_SDD_1576(PERF_SRD_2043, PERF_SRD_9751, PERF_SRD_2721)
128 137     PERF_SDD_1577(PERF_SRD_9751)
129 138     PERF_SDD_1578(PERF_SRD_9751, PERF_SRD_9752)
130 139     PERF_SDD_1580(PERF_SRD_9751)
131 140     PERF_SDD_1581_INT
132 141     PERF_SDD_3571(PERF_SRD_7473, PERF_SRD_9646)
133 142
134 143
135 144 INPUT                                                                                                  VALUE
136 145 ------------------------------------------------------------------------------------------------------  ------------------
      » --------
137 146 Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_is_valid
      »     true
138 147 Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_Data
      »     2370
139 148 Ctp_Perf_Despath_Calc_CC_Rate.Crzalt
      »  22050.0
140 149 Ctp_Perf_Despath_Calc_CC_Rate.Et
      »     35.0
141 150 Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq
      »      0.0
142 151 Ctp_Perf_Despath_Calc_CC_Rate.Critcralt
      »      0.0
143 152 Perf_Background_Dpkg.Psdestalt
      »    200.0
144 153 Perf_Background_Dpkg.Psdestisadev
      »      2.0
145 154 Perf_Background_Dpkg.Pstropoalt
      »  36089.0
146 155 Perf_Background_Dpkg.Pcsttary.Stt.Isadev
      »      2.0
147 156 Perf_Background_Dpkg.Pscabinrate
      »     29.0
148 157 Perf_Background_Dpkg.Psdestqnh.Data
      »   1013.0
149 158
150 159
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rst (continued)

151	160	OUTPUT	EXPECTED	TOLERANCE	ACTUAL	
		» P/F				
152	161	----------	----------	----------	----------	
		» -------- ----				
153	162	Pdb_Constants.Hreps01	0.872	0.001	8.7	
		» 2000E-01 P				
154	163	Pdb_Constants.Hreps02	597.0	0.001	5.9	
		» 7000E+02 P				
155	164	Pdb_Constants.Hreps03	7.000	0.001	7.0	
		» 0000E+00 P				
156	165	Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq	0.0	0.001	0.0	
		» 0000E+00 P				
157	166	Ctp_Perf_Despath_Calc_CC_Rate.Critcralt	2.20600E+04	0.001	2.2	
		» 0600E+04 P				
158	167					
159	168					
160	169	====> All 5 Comparisons Passed <====				
161	170					
162	171					
163	172	TESTID: 3				
164	173					
165	174	Et is such that Rptimereq is equal to 0.				
166	175	PERF_SDD_1576(PERF_SRD_2043, PERF_SRD_9751, PERF_SRD_2721)				
167	176	PERF_SDD_1577(PERF_SRD_9751)				
168	177	PERF_SDD_1578(PERF_SRD_9751, PERF_SRD_9752)				
169	178	PERF_SDD_1579_INT				
170	179	PERF_SDD_3571(PERF_SRD_7473, PERF_SRD_9646)				
171	180					
172	181					
173	182	INPUT			VALUE	
174	183	----------			----------	
		» --------				
175	184	Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_is_valid				
		» true				
176	185	Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_Data				
		» 2370				
177	186	Ctp_Perf_Despath_Calc_CC_Rate.Crzalt				
		» 20000.0				
178	187	Ctp_Perf_Despath_Calc_CC_Rate.Et				
		» 20.000				
179	188	Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq				
		» 0.0				
180	189	Ctp_Perf_Despath_Calc_CC_Rate.Critcralt				

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rst (continued)

```
                » 0.0
181  190 Perf_Background_Dpkg.Psdestalt
                »  200.0
182  191 Perf_Background_Dpkg.Psdestisadev
                » 2.0
183  192 Perf_Background_Dpkg.Pstropoalt
                » 36089.0
184  193 Perf_Background_Dpkg.Pcsttary.Stt.Isadev
                » 2.0
185  194 Perf_Background_Dpkg.Pscabinrate
                » 5.0
186  195 Perf_Background_Dpkg.Psdestqnh.Data
                »  450.0
187  196
188  197
189  198 OUTPUT                                                    EXPECTED              TOLERANCE              ACTUAL
                »           P/F
190  199 ---------------------------------------------------  ------------------------------  -------------  ------------------
                » --------  ----
191  200 Pdb_Constants.Hreps01                                                    0.872          0.001                  8.7
                » 2000E-01  P
192  201 Pdb_Constants.Hreps02                                                    597.0          0.001                  5.9
                » 7000E+02  P
193  202 Pdb_Constants.Hreps03                                                    7.000          0.001                  7.0
                » 0000E+00  P
194  203 Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq                                    0.0          0.001                  0.0
                » 0000E+00  P
195  204 Ctp_Perf_Despath_Calc_CC_Rate.Critcralt                                20010.0          0.001                  2.0
                » 0100E+04  P
196  205
197  206
198  207 ====> All 5 Comparisons Passed <====
199  208
200  209
201  210 TESTID: 4
202  211
203  212     Cc_Rate is calculated and repressurization is needed.  Destpress, Deltapress, and Stdcrzpress are all calculated c
                » orrectly
204  213     but not able to be looked at due to debugger optimizations.  Are confirmed from end results.  Same as A but differ
                » ent values
205  214     to confirm calculations.
206  215     PERF_SDD_1576(PERF_SRD_2043, PERF_SRD_9751, PERF_SRD_2721)
207  216     PERF_SDD_1577(PERF_SRD_9751)
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rst (continued)

```
208 | 217 |     PERF_SDD_1578(PERF_SRD_9751, PERF_SRD_9752)
209 | 218 |     PERF_SDD_1580(PERF_SRD_9751)
210 | 219 |     PERF_SDD_3571(PERF_SRD_7473, PERF_SRD_9646)
211 | 220 |
212 | 221 |
213 | 222 | INPUT                                                                                                  VALUE
214 | 223 | ------------------------------------------------------------------------------------------------  ------------------
    |     | » --------
215 | 224 | Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_is_valid
    |     | »     true
216 | 225 | Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_Data
    |     | »     2370
217 | 226 | Ctp_Perf_Despath_Calc_CC_Rate.Crzalt
    |     | »  20000.0
218 | 227 | Ctp_Perf_Despath_Calc_CC_Rate.Rvs
    |     | »      0.0
219 | 228 | Ctp_Perf_Despath_Calc_CC_Rate.Et
    |     | »     0.05
220 | 229 | Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq
    |     | »      0.0
221 | 230 | Ctp_Perf_Despath_Calc_CC_Rate.Critcralt
    |     | »      0.0
222 | 231 | Perf_Background_Dpkg.Psdestalt
    |     | »    200.0
223 | 232 | Perf_Background_Dpkg.Psdestisadev
    |     | »      2.1
224 | 233 | Perf_Background_Dpkg.Pstropoalt
    |     | »  36089.0
225 | 234 | Perf_Background_Dpkg.Pcsttary.Stt.Isadev
    |     | »      2.0
226 | 235 | Perf_Background_Dpkg.Pscabinrate
    |     | »    400.0
227 | 236 | Perf_Background_Dpkg.Psdestqnh.Data
    |     | »   1100.0
228 | 237 | ctp_perf_Despath_calc_cc_rate.Rvs
    |     | »      rvs
229 | 238 |
230 | 239 |
231 | 240 | OUTPUT                                                EXPECTED              TOLERANCE              ACTUAL
    |     | »           P/F
232 | 241 | ------------------------------------------------  ----------------------------  -------------  ------------------
    |     | » --------  ----
233 | 242 | Pdb_Constants.Hreps01                                                0.872          0.001                8.7
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rst (continued)

```
» 2000E-01  P
Pdb_Constants.Hreps02                                          597.0          0.001             5.9
» 7000E+02  P
Pdb_Constants.Hreps03                                          7.000          0.001             7.0
» 0000E+00  P
Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq                        7.039          0.001             7.0
» 3985E+00  P
Ctp_Perf_Despath_Calc_CC_Rate.Critcralt                  8.73624E+03          0.001             8.7
» 3624E+03  P
Ctp_Perf_Despath_Calc_CC_Rate.Rvs                             1600.0          0.001             1.6
» 0000E+03  P


====> All 6 Comparisons Passed <====


TESTID: 5

    RVS is limited to 2000 ft/min.
    PERF_SRD_9751


INPUT                                                                                          VALUE
-------------------------------------------------------------------------------------------  ------------------
» --------
Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_is_valid
»     true
Io_analog_dpkg:body.aero_engine_Ident_Key:body.public_Data
»     2370
Ctp_Perf_Despath_Calc_CC_Rate.Crzalt
»  20000.0
Ctp_Perf_Despath_Calc_CC_Rate.Rvs
»      0.0
Ctp_Perf_Despath_Calc_CC_Rate.Et
»     0.05
Ctp_Perf_Despath_Calc_CC_Rate.Rptimereq
»      0.0
Ctp_Perf_Despath_Calc_CC_Rate.Critcralt
»      0.0
Perf_Background_Dpkg.Psdestalt
»    200.0
Perf_Background_Dpkg.Psdestisadev
»      2.1
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rst (continued)

```
261  270 Perf_Background_Dpkg.Pstropoalt
         »  36089.0
262  271 Perf_Background_Dpkg.Pcsttary.Stt.Isadev
         »      2.0
263  272 Perf_Background_Dpkg.Pscabinrate
         »    600.0
264  273 Perf_Background_Dpkg.Psdestqnh.Data
         »   1100.0
265  274 ctp_perf_Despath_calc_cc_rate.Rvs
         »      rvs
266  275
267  276
268  277 OUTPUT                                                EXPECTED               TOLERANCE               ACTUAL
         »           P/F
269  278 ------------------------------------------------  ----------------------------  -------------  ------------------
         » --------  ----
270  279 Pdb_Constants.Hreps01                                                    0.872          0.001                  8.7
         » 2000E-01  P
271  280 Pdb_Constants.Hreps02                                                    597.0          0.001                  5.9
         » 7000E+02  P
272  281 Pdb_Constants.Hreps03                                                    7.000          0.001                  7.0
         » 0000E+00  P
273  282 Ctp_Perf_Despath_Calc_CC_Rate.Rvs                                       2000.0          0.001                  2.0
         » 0000E+03  P
274  283
275  284
276  285 ====> All 4 Comparisons Passed <====
277  286
278  287
279       Test End Time: Sep 02 15:12:11 2010
     288 Test End Time: Oct 22 13:28:07 2014
280  289 Test Generation System (TGS) Version v4.5.2, ps4082887-103
281  290 Current Program Library
282       d:\airbus_io_builds\a340\s1a063\lib\a29_cert_system.alb (root)
283       d:\airbus_io_builds\a340\s1a063\lib\csw_abpeg_006.alb
284       d:\airbus_io_builds\a340\s1a063\lib\mtyp.alb
285       d:\airbus_io_builds\a340\s1a063\lib\iotbx.alb
286       d:\airbus_io_builds\a340\s1a063\lib\medu.alb
287       d:\airbus_io_builds\a340\s1a063\lib\tou.alb
288       d:\airbus_io_builds\a340\s1a063\lib\nam.alb
289       d:\airbus_io_builds\a340\s1a063\lib\ops.alb
290       d:\airbus_io_builds\a340\s1a063\lib\bsve.alb
291       d:\airbus_io_builds\a340\s1a063\lib\opc.alb
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.rst (continued)

292		~~d:\airbus_io_builds\a340\s1a063\lib\io.alb~~
293		~~d:\airbus_io_builds\a340\s1a063\lib\isb.alb~~
294		~~d:\airbus_io_builds\a340\s1a063\lib\prnt.alb~~
295		~~d:\airbus_io_builds\a340\s1a063\lib\w429.alb~~
296		~~d:\airbus_io_builds\a340\s1a063\lib\com.alb~~
297		~~d:\airbus_io_builds\a340\s1a063\lib\fm.alb~~
298		~~D:\Airbus_IO_Builds\A340\S1A063\lib\fm2.alb~~
299		~~D:\A340_063\CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE\fm2_p.alb~~
300		~~D:\A340_063\CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE\my_Fm2_lib.alb~~
	291	c:\a340\builds\st2099\bld_st2099\libraries\a29_cert_system.alb (root)
	292	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\CSW_ABPEG_006.ALB
	293	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\mtyp.ALB
	294	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\Iotbx.alb
	295	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\mcdu.alb
	296	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\Tou.alb
	297	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\Nam.alb
	298	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\Ops.alb
	299	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\Bsvc.alb
	300	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\Opc.alb
	301	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\Io.alb
	302	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\Isb.alb
	303	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\prnt.alb
	304	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\w429.alb
	305	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\Com.alb
	306	C:\A340\Builds\ST2099\BLD_ST2099\Libraries\Fm.alb
	307	C:\a340\Builds\st2099\BLD_st2099\Libraries\fm2.alb
	308	C:\TEST_A340\CTP\update\CR2\CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE\new\fm2_p.alb
	309	C:\TEST_A340\CTP\update\CR2\CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE\new\my_fm2.alb

Mode: All Lines

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.bat

1	1	ECHO OFF
2	2	REM
3	3	REM BAT File
4	4	REM
5	5	REM CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.bat
6	6	REM
7	7	REM CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE Started Execution
8	8	ECHO ON
9	9	ECHO Building Library
10	10	%build_lib% A340 %test% fm2
11	11	ECHO Compiling Drv
12	12	%acomp% CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_D.ADA
13	13	%acomp% CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB
14	14	%ccomp% CTP_A340S1A_PDB_COMMON_OBJECTS.c
15	15	ECHO recompiling
16	16	%recomp%
17	17	ECHO Linking
18	18	%alink% CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_d
19	19	ECHO Running
20	20	%runtgs% CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE Y
21	21	ECHO CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE Completed Execution

Mode: All Lines

File: recompile.bat

1	1	rem recompile
2	2	rem Last modified by Deepak 10/07/03
3	3	
4	4	a29_recompile/noall_units/progress/nokeep/noexecute/config=%a29_config%/scope=global
5	5	echo off
6	6	%IO_CMDS_DIR%\replace ACMREF: d:\Airbus_IO_Builds\A340\%test%\cfg\
7	7	%IO_CMDS_DIR%\replace IO_INCLUDESDIR d:\Airbus_IO_Builds\A340\%test%\cfg\
8	8	%IO_CMDS_DIR%\replace INCLUDE_DIR: ""
9	9	%IO_CMDS_DIR%\replace A340_IOTM_A29_CONFIG.TXT %a29_config%
10	10	
11	11	%IO_CMDS_DIR%\REPLACE NOCHECKS NOCHECK
12	12	%IO_CMDS_DIR%\REPLACE PROG/LIST NOLIST/OPT=(ALL,NODEAD)
13	13	
14	14	if /I %a29_config% == My_nav_lib_cfg.txt %IO_CMDS_DIR%\REPLACE A340_FM_A29_CONFIG.TXT A340_NAV_A29_CONFIG.TXT
15	15	if /I %a29_config% == My_Biu_lib_cfg.txt %IO_CMDS_DIR%\REPLACE A340_FM_A29_CONFIG.TXT A340_BIU_A29_CONFIG.TXT
16	16	if /I %a29_config% == My_ioci_lib_cfg.txt %IO_CMDS_DIR%\REPLACE A340_FM_A29_CONFIG.TXT A340_IOCI_A29_CONFIG.TXT
17	17	if /I %a29_config% == My_Maint_lib_cfg.txt %IO_CMDS_DIR%\REPLACE A340_FM_A29_CONFIG.TXT A340_MAINT_A29_CONFIG.TXT
18	18	if /I %a29_config% == My_fm2_cfg.txt %IO_CMDS_DIR%\REPLACE A340_FM_A29_CONFIG.TXT A340_FM2_A29_CONFIG.TXT
19	19	
20	20	%IO_CMDS_DIR%\replace A340_FM_A29_CONFIG.TXT My_fm2_lib_cfg.txt
21	21	%IO_CMDS_DIR%\replace A340_NAV_A29_CONFIG.TXT My_nav_lib_cfg.txt
22	22	%IO_CMDS_DIR%\replace A340_BIU_A29_CONFIG.TXT My_biu_lib_cfg.txt
23	23	%IO_CMDS_DIR%\replace A340_IOCI_A29_CONFIG.TXT My_ioci_lib_cfg.txt
24	24	%IO_CMDS_DIR%\replace A340_MAINT_A29_CONFIG.TXT My_maint_lib_cfg.txt
25	25	
26	26	echo on
27	27	recomp

Mode: All Lines

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_D.ADA

```
 1  1 --
 2  2 --    A340S1A COMPONENT TEST DRIVER
 3  3 --
 4  4 --    COMPONENT : Ctp_A340S1A_PERF_DESPATH_CALC_CC_RATE_D.ADA
 5  5 --
 6  6
 7  7
 8  8 with Portable_Types_Pkg;
 9  9
10 10 --
11 11 package CTP_PERF_DESPATH_CALC_CC_RATE is
12 12 -- Global test variables go here
13 13 --
14 14   Crzalt               : Portable_Types_Pkg.Float_32;
15 15   Et                  : Portable_Types_Pkg.Float_32;
16 16   Rptimereq           : Portable_Types_Pkg.Float_32;
17 17   Critcralt           : Portable_Types_Pkg.Float_32;
18 18   Destpress           : Portable_Types_Pkg.Float_32;
19 19   Stdcrzpress         : Portable_Types_Pkg.Float_32;
20 20   Tempratio           : Portable_Types_Pkg.Float_32;
21 21   Thetadev            : Portable_Types_Pkg.Float_32;
22 22   Pressratio          : Portable_Types_Pkg.Float_32;
23 23   Rvs                 : Portable_Types_Pkg.Float_32;
24 24
25 25   Aero_engine_key : Portable_Types_Pkg.Integer_32;   --newly added
26 26
27 27
28 28 end CTP_PERF_DESPATH_CALC_CC_RATE;
29 29 --
30 30 --
31 31 --
32 32 with CTP_PERF_DESPATH_CALC_CC_RATE;
33 33 use CTP_PERF_DESPATH_CALC_CC_RATE;
34 34 with Apex_Partition_Pkg;
35 35 use Apex_Partition_Pkg;
36 36 with Perf_Pdb_Initialization_Pkg;
37 37 with Perf_Initialization_Pkg;
38 38 with Prf_Despath_Pkg;
39 39 use Prf_Despath_pkg;
40 40
41 41 procedure CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_d is
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_D.ADA (continued)

```
42  42
43  43  Boot_Status        : Apex_Partition_Pkg.Operating_Mode_Type := Normal;
44  44  Config_Found :   Boolean :=False;
45  45  No_Aedb_Loaded : Boolean :=True;
46  46  Signature_Fail : Boolean :=True;
47  47  Version_Compat : Boolean :=False;
48  48
49  49  begin
50  50
51  51  -- Initialize Perf database
52  52    Perf_Initialization_Pkg.Initialize (Boot_Status);
53  53    Perf_Pdb_Initialization_Pkg.Init_Pdb (Config_Found, No_Aedb_loaded, Signature_Fail, Version_Compat);
54  54
55  55
56  56    --  execute SUT
57  57    Calc_Cc_Rate (Crzalt, Et, Rptimereq, Critcralt);
58  58    <<testend>>  NULL;
59  59  end CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE_d;
```

Mode: All Lines

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.CUL

1	1	##
2	2	## CUL FILE
3	3	##
4	4	## CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.CUL
5	5	##
6	6	PRF_DESPATH_PKG.CALC_CC_RATE
7	7	##

Mode: All Lines

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.dsp

```
1  1  ##*****************************************************************************************************************************
      » **
2  2  ##    DSP Generator Tool Version 1.0
3  3  ##*****************************************************************************************************************************
      » **
4  4  ##
5  5  ##    CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.DSP
6  6  ##
7  7  ## NOTE:
8  8  ## A. "Any" SCR that is mentioned in this DSP file must contain the prefix "SCR_disposed#: "
9  9  ## B. Template of this DSP file is created by tool and it should not be modified/deleted.
10 10 ## C. If any information is not applicable then mark the corresponding field as N/A instead of deleting it.
11 11 ## D. If more than one SCR has to be used for one issue, make separate entry. SCRs should not be captured
12 12 ##    in the same line using comma or any other separators.
13 13 ##
14 14 ##
15 15
16 16 ---------------------------------------------------------------------------------------------------------------
      » --
17 17 1. REASON_FOR_FAILURES_OF_TEST_CASE(S):
18 18 ## The below mentioned group of lines need to be repeated for each Test case ID, which is having test failures in it.
19 19 ---------------------------------------------------------------------------------------------------------------
      » --
20 20
21 21 Test_case_Id: N/A
22 22 #_of_Failures: N/A
23 23 Failed_Requirements: N/A
24 24 SCR_disposed#: N/A
25 25 SCR_PROJECT: N/A
26 26 SCR_SUB_PROJECT: N/A
27 27 Disposition: N/A
28 28
29 29 ---------------------------------------------------------------------------------------------------------------
      » --
30 30 2. COVERAGE_PROBLEM(S):
31 31 ## Standard excuse and SCR related details need to be mentioned for each and every sub unit separately.
32 32 ---------------------------------------------------------------------------------------------------------------
      » --
33 33 Compilation_Unit_Name: PRF_DESPATH_PKG.CALC_CC_RATE
34    ~~Uncovered_Code:~~
   34 Uncovered_Code: N/A
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.dsp (continued)

```
35  35
36  36
37  37  TCH(Test_Coverage_Hole)_Excuse: N/A
38  38  N/A
39  39  SCR_disposed#: N/A
40  40  SCR_PROJECT: N/A
41  41  SCR_SUB_PROJECT: N/A
    42
42  43
43  44  ------------------------------------------------------------------------------------------------------------------------
        » --
44  45  3. ANY_OTHER_ISSUE(S):
45  46  ## A. Every entry in Any_Other_Issue should be followed by a SCR_number, its corresponding CM 21 project and subprojec
        » t.
46  47  ## B. If SCR is not applicable then mention N/A.
47  48  ## C. If more than one SCR has to be used for one issue, make separate entry. SCRs should not be captured
48  49  ##    in the same line using comma or any other separators.
49  50  ------------------------------------------------------------------------------------------------------------------------
        » --
50  51
51  52  (i) Anchors Perf_sdd_3571 and perf_srd_7473 could not be tested explicitly. As values
52  53      of the pdb constants are passed directly. It is tested implicitly from the end results
53  54      and values of the pdb constants checked for all the test cases.
54  55  SCR_disposed#: N/A
55  56  SCR_PROJECT: N/A
56  57  SCR_SUB_PROJECT: N/A
    58
57  59
58  60  ------------------------------------------------------------------------------------------------------------------------
        » --
59  61  4. SPECIAL_EXECUTION_INSTRUCTION(S):
60  62  ## Capture all additional information and/or supporting file(s) required for this CTP execution.
61  63  ## For example:
62  64  ## (i) "nav_db23.o" is required for execution.
63  65  ## (ii) "apex_traps.o"/gen=xx and "common file"/gen=xx are required for execution.
64  66  ## Database_Details:
65  67  ## 1. <Enter the database name>
66  68  ------------------------------------------------------------------------------------------------------------------------
        » --
67  69
68  70  APEX_TRAPS.O/gen=3 & CTP_A340S1A_PDB_COMMON_OBJECTS.C/gen=1 are required for execution.
69  71
70  72  Database_Details:
```

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.dsp (continued)

71	73	1. PDB_PS203C-00283_88780000.COFF
72	74	
73	75	********** End of Report **********
		» **

Mode: All Lines

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB

```
--!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
--!
--!    STUB File
--!
--!    Ctp_A340S1A_Perf_Cs_Db_Acces_Gnrc.Stb
--!
--!    REASONS FOR STUBBING : This Stub contains routines for accessing information from
--!    the header of databases which follow the 777 AIMS Common Database Format
--!    (reference Honeywell CM AIMS-H-01320), database checks, and a package containing
--!    routines to access database files.
--!
--!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

with Cs_Database_Access_Iftypes;
use Cs_Database_Access_Iftypes;
with Portable_Types_Pkg;
use Portable_Types_Pkg;
with Unchecked_Conversion;
with System;
with Apex_Types_Pkg;
use Apex_Types_Pkg;


package body Cs_Database_Access_Gnrc is

--!
--  PURPOSE:     This package contains routines for accessing information from the header of databases which follow th
» e 777
--               AIMS Common Database Format (reference Honeywell CM AIMS-H-01320), database checks, and a package con
» taining
--               routines to access database files.
--  DATA_RIGHTS: Honeywell ATSD Proprietary
--  ANCHOR:      CS_CODE_1755
--  RAISES:      N/A
--  SPECIAL_CONSIDERATIONS:
--      This package is instantiated with the starting address of the data base.
--
--  REVISION_HISTORY:
--    Date    SCR/SPR  Author               Description of change
--    ======  =======  ===================  ========================
--    032693    88.00  K. Mason             Initial Release
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
40  40 --    081193   227.00  LS Warner            Updates per parameter changes
41  41 --                                          for Database_Is_Loaded, withed
42  42 --    112993   270.00  S. LaPlante          Added routine to access by
43  43 --                                          Byte offset
44  44 --    011494   274.00  S. Checinski         Added routine to access next
45  45 --                                          fixed length record in file
46  46 --                                          in Load_Map_Common_Types_Pkg
47  47 --    012594   297.00  M. Bloch             Added the declaration of
48  48 --                                          load_map_indicators_db_sram and
49  49 --                                          the associated address clause
50  50 --    041395   475.00  T.S.Manecke          Prevented read_through procedure
51  51 --                                          from being optimized away
52  52 --    051795   598.00  Kevin Tucker         Added IF stmnt to File_Exists to check
53  53 --                                          whether the requested file # even exists.
54  54 --                                          Also moved File_Pointer_Record from being a package global
55  55 --                                          to CS_File_Access_Gnrc to a local of Initialize_File.
56  56 --                                         Finally, made Initialize_File default output data if asked
57  57 --                                         to work on file whose FPR isn't present.
58  58 --    ======  Honeywell Common Software Project ======
59  59 --    110995    41.00  S. Darr              Initial release in Honeywell
60  60 --                                          Common Software ACM project
61  61 --                                          with the new generic
62  62 --                                          formal parameter profile.
63  63 --    060796  VIA 927.00  Ajit Prem      Type definitions and related
64  64 --        CSW 76.00              routines have been modified to
65  65 --                                          prevent non-word data accesses to
66  66 --                                          a database.
67  67 --    040398  CSW 151.00  Tom Williams      Moved initialization of
68  68 --                                          Database_Header.Version to inside
69  69 --                                          IF statement.
70  70 --
71  71 --!
72  72 --
73  73     Word : constant := 4;
74  74 --      size of a word for use in representation clauses
75  75     Bytes_Per_Word : constant := 4;
76  76 --      used to change an offset given in 32-bit words to byte addressing
77  77
78  78
79  79   type Creation_Date_Type is record
80  80     Spare           : Portable_Types_Pkg.Unsigned_11;
81  81     Year            : Portable_Types_Pkg.Unsigned_12;
82  82     Month           : Portable_Types_Pkg.Unsigned_4;
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
    Day             : Portable_Types_Pkg.Unsigned_5;
  end record;

  for Creation_Date_Type use record
    Spare at 0 range 0..10;
    Year  at 0 range 11..22;
    Month at 0 range 23..26;
    Day   at 0 range 27..31;
  end record;

  type Fpr_Attributes_Type is record
    Spare           : Portable_Types_Pkg.Unsigned_8;
    Fpr_Size        : Portable_Types_Pkg.Unsigned_8;    -- FPR size in words
    Num_Of_Fprs     : Portable_Types_Pkg.Unsigned_16;   -- number of FPRS in database
  end record;

  for Fpr_Attributes_Type use record
    Spare          at 0 range 0..7;
    Fpr_Size       at 0 range 8..15;
    Num_Of_Fprs    at 0 range 16..31;
  end record;


    type Database_Header_Type is
    record
        Load_Complete : Portable_Types_Pkg.Integer_32;    -- not used
        Header_Size :
           Cs_Database_Access_Iftypes.
          Database_Size_Type;           -- size in 32 bit words of database header
            Version     : Cs_Database_Access_Iftypes.Database_Version_Type;
        Database_Size :
           Cs_Database_Access_Iftypes.
          Database_Size_Type;  -- size in 32 bit words of database including header
                       -- and data
        Database_Crc : Portable_Types_Pkg.
                  Unsigned_32;   -- Database CRC
            Creation_Date  : Creation_Date_Type;
        First_Fpr_Address :
           Cs_Database_Access_Iftypes.
          Db_Memory_Address_Type;           -- Pointer to first File Pointer Record
            Fpr_Attributes : Fpr_Attributes_Type;
        Database_Signature :
           Cs_Database_Access_Iftypes.
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
126 126           Database_Signature_Type;          -- Kind of database (i.e. FMCF_AMI)
127 127        Database_Identifier :
128 128           Cs_Database_Access_Iftypes.
129 129           Identifier_Type;                 -- Specific database (i.e. UAL0001)
130 130    end record;
131 131
132 132    for Database_Header_Type use
133 133    record
134 134        Load_Complete       at 0 * Word range 0 .. 31;
135 135        Header_Size         at 1 * Word range 0 .. 31;
136 136        Version             at 2 * Word range 0 .. 31;
137 137        Database_Size       at 3 * Word range 0 .. 31;
138 138        Database_Crc        at 4 * Word range 0 .. 31;
139 139            Creation_Date       at 5 * Word range 0 .. 31;
140 140        First_Fpr_Address   at 6 * Word range 0 .. 31;
141 141        Fpr_Attributes      at 7 * Word range 0 .. 31;
142 142        Database_Signature  at 8 * Word range 0 .. 63;
143 143        Database_Identifier at 10 * Word range 0 .. 127;
144 144    end record;
145 145
146 146    type Access_To_Database_Header_Type is access Database_Header_Type;
147 147
148 148 --         Data base header, follows CM AIMS-H-01320
149 149
150 150    function To_Access_To_Database_Header_Type is
151 151       new Unchecked_Conversion (Source => Portable_Types_Pkg.Unsigned_32,
152 152                 Target => Access_To_Database_Header_Type);
153 153 --         Convert from instantiated address to access type
154 154
155 155    Database_Header : Access_To_Database_Header_Type;
156 156    Start_Address : Portable_Types_Pkg.Unsigned_32;
157 157    Loaded : Boolean;
158 158
159 159 --  Access to the database header, global to this package, set up by procedure Initialize_Header
160 160
161 161    type First_Word_Type is record
162 162      File_Exists       : Boolean;    -- true => file exists in DB
163 163      Spare             : Portable_Types_Pkg.Integer_3;
164 164      Field_Enable_Bits : Portable_Types_Pkg.Integer_12;    -- Field enable bits
165 165      Record_Size       : Unsigned_16;
166 166    end record;
167 167
168 168    for First_Word_Type use record
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
169 169       File_Exists       at 0 range 0..0;
170 170       Spare             at 0 range 1..3;
171 171       Field_Enable_Bits at 0 range 4..15;
172 172       Record_Size       at 0 range 16..31;
173 173     end record;
174 174
175 175     type File_Pointer_Record_Type is
176 176     record
177 177             First_Word  : First_Word_Type;
178 178         Pointer_To_File :
179 179            Cs_Database_Access_Iftypes.
180 180           Db_Memory_Address_Type;    -- Offset in 32 bit words from first
181 181                         -- word in the database to the first word
182 182                         -- in the file
183 183         File_Size_Or_Rec_Count :
184 184            Cs_Database_Access_Iftypes.
185 185           File_Size_Type;        -- Number of records if record_size > 0 or
186 186                       -- Number of 32 bit words if record_size = 0
187 187     end record;
188 188     for File_Pointer_Record_Type use
189 189     record
190 190             First_Word  at 0 * Word range 0..31;
191 191         Pointer_To_File at 1 * Word range 0 .. 31;
192 192         File_Size_Or_Rec_Count at 2 * Word range 0 .. 31;
193 193     end record;
194 194
195 195     type Access_To_File_Pointer_Record_Type is access File_Pointer_Record_Type;
196 196
197 197
198 198 --          File Pointer Record, follows CM AIMS-H-01320
199 199
200 200     function To_Access_To_File_Pointer_Record_Type is
201 201        new Unchecked_Conversion
202 202           (Source => Cs_Database_Access_Iftypes.Db_Memory_Address_Type,
203 203            Target => Access_To_File_Pointer_Record_Type);
204 204 --          Convert from address in header to access type
205 205
206 206     type Word_Type is new Portable_Types_Pkg.Integer_32;
207 207     Temp_Word : Word_Type;  -- Objest to read database memory
208 208
209 209     procedure Initialize_Header is
210 210 --!
211 211 --      DESCRIPTION:     This procedure will set up an object of access type to the database header.
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
--
--      DATA_RIGHTS: Honeywell ATSD Proprietary
--      ANCHOR: CS_CODE_1756
--      SHARED_DATA:
--               Name                                                    Mode
--              ================================================================
--              File_Id_Value                                           In
--                The parameter used to instantiate generic package CS_Database_Access_GNRC
--              Database_Start_Address                                  In
--                The parameter used to instantiate generic package CS_Database_Access_GNRC
--              Database_Header                                         Out
--                An access type to the database header
--
--      SPECIAL_CONSIDERATIONS: N/A
--!

    Starting_Address : System.address;
    Starting_Physical_Address : System.Address;
    Size : Portable_Types_Pkg.Natural_32;
  Actual_Size : Portable_Types_Pkg.Natural_32;
  CRC  : Portable_Types_Pkg.Integer_32;
  Part_Number : Apex_Types_Pkg.String_16_Type;

    Status : Apex_Types_Pkg.Status_Code_Type;
-- --| @DESCRIPTION Beginning address of the OPC database
-- --| for PMT:  88980000 below is its two's compliment
--   Opc_Address_Const_t : constant := -16#77680000#;
--   Opc_Address_Const : constant Portable_Types_Pkg.Address_Zero := Opc_Address_Const_t;
--
-- --| @DESCRIPTION Beginning address of the AMI database
-- --| for PMT:  88A00000 below is its two's compliment
--   Ami_Address_Const_t : constant := -16#77600000#;
--   Ami_Address_Const : constant Portable_Types_Pkg.Address_Zero := Ami_Address_Const_t;
--
-- --| @DESCRIPTION Beginning address of the AEDB database
-- --| for PMT:  88780000 below is its two's compliment
--   Aedb_Address_Const_t : constant := -16#77880000#;
--   Aedb_Address_Const : constant Portable_Types_Pkg.Address_Zero := Aedb_Address_Const_t;

--| @DESCRIPTION Beginning address of the OPC database
--| for PMT:  88780000 below is its two's compliment
  Opc_Address_Const_t : constant := -16#77780000#;
  Opc_Address_Const : constant Portable_Types_Pkg.Unsigned_32 := Opc_Address_Const_t;
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
255 255
256 256 --| @DESCRIPTION Beginning address of the AMI database
257 257 --| for PMT:  88800000 below is its two's compliment
258 258   Ami_Address_Const_t : constant := -16#77780000#;
259 259   Ami_Address_Const : constant Portable_Types_Pkg.Unsigned_32 := Ami_Address_Const_t;
260 260
261 261 --| @DESCRIPTION Beginning address of the AEDB database
262 262 --| for PMT:  88880000 below is its two's compliment
263 263   Aedb_Address_Const_t : constant := -16#77880000#; --88780000
264 264   Aedb_Address_Const : constant Portable_Types_Pkg.Unsigned_32 := Aedb_Address_Const_t;
265 265
266 266     begin -- procedure Initialize_Header
267 267
268 268     case (Portable_Types_Pkg.Integer_32(File_Id_Value)) is
269 269
270 270       when 18 =>  -- AEDB
271 271         Start_Address := Aedb_Address_Const;
272 272         Loaded := True;
273 273
274 274       when 14 =>  -- AMI
275 275         Start_Address := Ami_Address_Const;
276 276         Loaded := True;
277 277
278 278       when 7 =>  -- OPC
279 279         Start_Address := Opc_Address_Const;
280 280         Loaded := True;
281 281
282 282       when others =>
283 283         Start_Address := 16#00000000#;
284 284         Loaded := False;
285 285
286 286     end case;
287 287
288 288
289 289
290 290 --    if (Portable_Types_Pkg.Integer_32 (File_Id_Value) /= 0) then
291 291 --      Apex_Extension_Pkg.Get_Database_Info (File_Id => Portable_Types_Pkg.Natural_32 (File_Id_Value),
292 292 --                                            Starting_Address => Starting_Address,
293 293 --                                            Allocated_Size => Size,
294 294 --                                            Is_Loaded => Loaded,
295 295 --                                            Status_Code => Status);
296 296 --      Start_Address := Starting_Address;
297 297 --    else
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
298 298 --      Start_Address := Portable_Types_Pkg.Unsigned_32;
299 299 --      Loaded := True;
300 300 --    end if;
301 301 --
302 302 --    if (Database_Start_Address /= Portable_Types_Pkg.Address_Zero) then
303 303 --      Start_Address := Database_Start_Address;
304 304 --    end if;
305 305 --
306 306     Database_Header := To_Access_To_Database_Header_Type (Start_Address);
307 307
308 308     end Initialize_Header;
309 309
310 310 --
311 311 --      2. Routines to perform checks
312 312 --
313 313     procedure Database_Ok (Database_Signature : in Cs_Database_Access_Iftypes.
314 314                                 Database_Signature_Type;
315 315                 Ops_Min_Db_Minor_Version : in
316 316                    Cs_Database_Access_Iftypes.Minor_Version_Type;
317 317                 Ops_Min_Db_Major_Version : in
318 318                    Cs_Database_Access_Iftypes.Major_Version_Type;
319 319                 Ops_Max_Db_Major_Version : in
320 320                    Cs_Database_Access_Iftypes.Major_Version_Type;
321 321                 Status : out Cs_Database_Access_Iftypes.
322 322                         Database_Check_Status_Type) is
323 323 --!
324 324 --          PURPOSE:     Verifies the database has been loaded, performs a signature field check, and
325 325 --                       performs a major/minor version compatibility test.
326 326 --
327 327 --          DESCRIPTION: The Loaded flag set during Initialize_Header is checked to determine if the
328 328 --                       database has been successfully loaded. If the database has been successfully loaded, Database
    » _signature
329 329 --                       is compared against the Data Base Signature read from the
330 330 --                       data base to ensure the correct type of database is stored at the base_address. OPS_DB_Versio
    » n is then
331 331 --                       compared to the Major Version and Minor Version stored in the database to ensure the operatio
    » nal program
332 332 --                       software and the database are compatible.
333 333 --          DATA_RIGHTS: Honeywell ATSD Proprietary
334 334 --          ANCHOR: CS_CODE_1757
335 335 --          SHARED_DATA:
336 336 --              Name                                                    Mode
337 337 --              ============================================================
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
338 338 --              Database_Header                                  In
339 339 --              An access type to the database header
340 340 --          SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
341 341 --!
342 342       Database_Version : Cs_Database_Access_Iftypes.Database_Version_Type;
343 343
344 344     begin -- Procedure Database_OK
345 345     if not Loaded then                                    -- Check if DB is loaded
346 346         Status := Not_Loaded;
347 347     elsif (Database_Signature /=
348 348            Database_Header.Database_Signature) then       -- Perform signature check
349 349         Status := Signature_Mismatch;
350 350     else
351 351             Database_Version := Database_Header.Version;
352 352             if (Ops_Min_Db_Major_Version >
353 353                Database_Version.Major_Version) or else    -- Perform check for minimum database version
354 354               ((Ops_Min_Db_Major_Version =
355 355             Database_Version.Major_Version) and then
356 356                (Ops_Min_Db_Minor_Version >
357 357             Database_Version.Minor_Version)) or else
358 358               (Ops_Max_Db_Major_Version <
359 359                Database_Version.Major_Version) then       -- and check for maximum database version
360 360             Status := Incompatible_Version;
361 361         else                             -- if the above passed, all is okay
362 362             Status := Okay;
363 363             end if;
364 364     end if;
365 365     end Database_Ok;
366 366
367 367
368 368 function Database_Is_Loaded
369 369   (
370 370    File_Identifier : in Cs_Database_Access_Iftypes.File_Identifier_Type
371 371   )
372 372 return boolean is
373 373
374 374 --!
375 375 -- DATA_RIGHTS:  Honeywell ATSD Proprietary
376 376 -- ANCHOR:       CS_CODE_1758
377 377 -- DESCRIPTION:  This function uses the file id passed in as a parameter
378 378 --               to call the APEX interface Get_Database_Info to find the
379 379 --               load Status of the Database.
380 380 -- SHARED_DATA:
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
381 381 --   N/A
382 382 --
383 383 -- SPECIAL_CONSIDERATIONS:
384 384 --  N/A
385 385 --
386 386 --!
387 387
388 388     --Starting_Address : Portable_Types_Pkg.Unsigned_32;
389 389       --Size : Portable_Types_Pkg.Natural_32;
390 390     --Status : Apex_Types_Pkg.Status_Code_Type;
391 391         Loaded : Boolean;
392 392     Starting_Physical_Address : System.Address;
393 393   Actual_Size : Portable_Types_Pkg.Natural_32;
394 394   CRC  : Portable_Types_Pkg.Integer_32;
395 395   Part_Number : Apex_Types_Pkg.String_16_Type;
396 396
397 397     begin
398 398 --      Apex_Extension_Pkg.Get_Database_Info
399 399 --      (
400 400 --      File_Id => Portable_Types_Pkg.Natural_32 (File_Identifier),
401 401 --      Starting_Virtual_Address => Starting_Address,
402 402 --      Starting_Physical_Address => Starting_Physical_Address,
403 403 --      Allocated_Size => Size,
404 404 --      Actual_Size => Actual_Size,
405 405 --      CRC => CRC,
406 406 --      Part_Number => Part_Number,
407 407 --      Is_Loaded => Loaded,
408 408 --      Status_Code => Status);
409 409
410 410 --      if Status /= Apex_Types_Pkg.No_Error then
411 411 --        Loaded := False;
412 412 --     end if;
413 413     case (Portable_Types_Pkg.Natural_32(File_Identifier)) is
414 414
415 415       when 18 =>  -- AEDB
416 416         Loaded := True;
417 417
418 418       when 14 =>  -- AMI
419 419         Loaded := True;
420 420
421 421       when 7 =>  -- OPC
422 422         Loaded := True;
423 423
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
424 424       when others =>
425 425         Loaded := False;
426 426
427 427     end case;
428 428       return (Loaded);
429 429
430 430     end Database_Is_Loaded;
431 431
432 432
433 433     procedure Read_Through is
434 434 --!
435 435 --          DESCRIPTION: This procedure performs a read through of the database.
436 436 --          DATA_RIGHTS: Honeywell ATSD Proprietary
437 437 --          ANCHOR: CS_CODE_1759
438 438 --          SHARED_DATA:
439 439 --              Name                                                    Mode
440 440 --              ================================================================
441 441 --              Database_Header                                         In
442 442 --              An access type to the database header
443 443 --          SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
444 444 --!
445 445     type Access_Word_Type is access Word_Type;
446 446     function Database_Size_To_Access_Word_Type is
447 447        new Unchecked_Conversion
448 448           (Source => Cs_Database_Access_Iftypes.Database_Size_Type,
449 449            Target => Access_Word_Type);
450 450     Access_Word : Access_Word_Type;      -- Access to database memory
451 451     Database_End_Address : Cs_Database_Access_Iftypes.Database_Size_Type;
452 452
453 453     begin -- Procedure Read_Through
454 454
455 455     Database_End_Address := Start_Address + Database_Header.Database_Size;
456 456     for Word_To_Read in 0 ..
457 457                    ((Database_End_Address - 1) - Start_Address) loop
458 458         Access_Word := Database_Size_To_Access_Word_Type
459 459                     (Start_Address + Bytes_Per_Word * Word_To_Read);
460 460         Temp_Word := Access_Word.all;
461 461     end loop;
462 462     Temp_Word := Temp_Word + 1;
463 463
464 464     end Read_Through;
465 465 --
466 466 --      3. Routines to return data from the data base header.
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
467 467 --
468 468     function Record_Size (File_Number : in
469 469                  Cs_Database_Access_Iftypes.File_Num_Type)
470 470                return Cs_Database_Access_Iftypes.Record_Size_Type is
471 471 --!
472 472 --      DESCRIPTION:     This function returns the size of records for files which contain fixed length records.
473 473 --
474 474 --      DATA_RIGHTS: Honeywell ATSD Proprietary
475 475 --      ANCHOR: CS_CODE_1760
476 476 --          SHARED_DATA:
477 477 --              Name                                                    Mode
478 478 --              ========================================================================
479 479 --              start_address                                           In
480 480 --              Starting address of database
481 481 --              Database_Header                                         In
482 482 --              An access type to the database header
483 483 --      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
484 484 --!
485 485 --
486 486     Access_To_File_Pointer_Record : Access_To_File_Pointer_Record_Type;
487 487     File_Pointer_Start_Address :
488 488        Cs_Database_Access_Iftypes.Db_Memory_Address_Type;
489 489         Fpr_Attributes : Fpr_Attributes_Type;
490 490         First_Word     : First_Word_Type;
491 491     begin
492 492         Fpr_Attributes := Database_Header.Fpr_Attributes;
493 493     File_Pointer_Start_Address :=
494 494        Start_Address + Bytes_Per_Word *
495 495                    (Database_Header.First_Fpr_Address +
496 496                     Db_Memory_Address_Type (File_Number) *
497 497                    Cs_Database_Access_Iftypes.
498 498                       Db_Memory_Address_Type
499 499                        (Fpr_Attributes.Fpr_Size));
500 500     Access_To_File_Pointer_Record :=
501 501        To_Access_To_File_Pointer_Record_Type (File_Pointer_Start_Address);
502 502         First_Word := Access_To_File_Pointer_Record.First_Word;
503 503     return Cs_Database_Access_Iftypes.Record_Size_Type
504 504           (First_Word.Record_Size);
505 505     end Record_Size;
506 506
507 507     function File_Size_Or_Record_Count
508 508         (File_Number : in Cs_Database_Access_Iftypes.File_Num_Type)
509 509         return Cs_Database_Access_Iftypes.File_Size_Type is
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
510 | 510 |--!
511 | 511 |--      DESCRIPTION:    This function returns the number of records for a file containing fixed length records, or num
    |     |» ber of
512 | 512 |--                      (32-bit) data words in the file if the file contains variable length records.
513 | 513 |--
514 | 514 |--      DATA_RIGHTS: Honeywell ATSD Proprietary
515 | 515 |--      ANCHOR: CS_CODE_1761
516 | 516 |--      SHARED_DATA:
517 | 517 |--              Name                                                    Mode
518 | 518 |--              ================================================================
519 | 519 |--              start_address                                           In
520 | 520 |--              Starting address of database
521 | 521 |--              Database_Header                                         In
522 | 522 |--              An access type to the database header
523 | 523 |--      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
524 | 524 |--!
525 | 525 |    File_Pointer_Start_Address :
526 | 526 |       Cs_Database_Access_Iftypes.Db_Memory_Address_Type;
527 | 527 |    Access_To_File_Pointer_Record : Access_To_File_Pointer_Record_Type;
528 | 528 |        Fpr_Attributes : Fpr_Attributes_Type;
529 | 529 |    begin
530 | 530 |        Fpr_Attributes := Database_Header.Fpr_Attributes;
531 | 531 |    File_Pointer_Start_Address :=
532 | 532 |       Start_Address + Bytes_Per_Word *
533 | 533 |                   (Database_Header.First_Fpr_Address +
534 | 534 |                    Cs_Database_Access_Iftypes.Db_Memory_Address_Type
535 | 535 |                   (File_Number) *
536 | 536 |                    Cs_Database_Access_Iftypes.Db_Memory_Address_Type
537 | 537 |                   (Fpr_Attributes.Fpr_Size));
538 | 538 |    Access_To_File_Pointer_Record :=
539 | 539 |       To_Access_To_File_Pointer_Record_Type (File_Pointer_Start_Address);
540 | 540 |    return Access_To_File_Pointer_Record.File_Size_Or_Rec_Count;
541 | 541 |    end File_Size_Or_Record_Count;
542 | 542 |
543 | 543 |    function File_Exists
544 | 544 |        (File_Number : in Cs_Database_Access_Iftypes.File_Num_Type)
545 | 545 |        return Boolean is
546 | 546 |--!
547 | 547 |--      DESCRIPTION:    This determines whether or not a given file exists in the database.
548 | 548 |--
549 | 549 |--      DATA_RIGHTS: Honeywell ATSD Proprietary
550 | 550 |--      ANCHOR: CS_CODE_1762
551 | 551 |--      SHARED_DATA:
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
552 552 --              Name                                                    Mode
553 553 --              ========================================================================
554 554 --              start_address                                           In
555 555 --              Starting address of database
556 556 --              Database_Header                                         In
557 557 --              An access type to the database header
558 558 --      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
559 559 --!
560 560     File_Pointer_Start_Address :
561 561        Cs_Database_Access_Iftypes.Db_Memory_Address_Type;
562 562     Access_To_File_Pointer_Record : Access_To_File_Pointer_Record_Type;
563 563         Fpr_Attributes : Fpr_Attributes_Type;
564 564         First_Word     : First_Word_Type;
565 565     begin
566 566         Fpr_Attributes := Database_Header.Fpr_Attributes;
567 567
568 568     -- CSW SCR #598 File_Exists should return FALSE if asked about a file that doesn't
569 569     --              exist in the database (because its FPR isn't present).
570 570
571 571     if (Portable_Types_Pkg.Unsigned_16 (File_Number) >=
572 572         Fpr_Attributes.Num_Of_Fprs) then
573 573         return False;
574 574     end if;
575 575
576 576     File_Pointer_Start_Address :=
577 577        Start_Address + Bytes_Per_Word *
578 578                    (Database_Header.First_Fpr_Address +
579 579                     Cs_Database_Access_Iftypes.Db_Memory_Address_Type
580 580                    (File_Number) *
581 581                     Cs_Database_Access_Iftypes.Db_Memory_Address_Type
582 582                    (Fpr_Attributes.Fpr_Size));
583 583     Access_To_File_Pointer_Record :=
584 584        To_Access_To_File_Pointer_Record_Type (File_Pointer_Start_Address);
585 585         First_Word := Access_To_File_Pointer_Record.First_Word;
586 586     return First_Word.File_Exists;
587 587     end File_Exists;
588 588
589 589     function Field_Enable (File_Number : in
590 590                    Cs_Database_Access_Iftypes.File_Num_Type)
591 591                return Portable_Types_Pkg.Unsigned_32 is
592 592 --!
593 593 --      DESCRIPTION:    This function returns the Field Enable Bits for a given file.
594 594 --
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
595 595 --      DATA_RIGHTS: Honeywell ATSD Proprietary
596 596 --      ANCHOR: CS_CODE_1763
597 597 --      SHARED_DATA:
598 598 --              Name                                                    Mode
599 599 --              ================================================================
600 600 --              start_address                                           In
601 601 --              Starting address of database
602 602 --              Database_Header                                         In
603 603 --              An access type to the database header
604 604 --      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
605 605 --!
606 606     File_Pointer_Start_Address :
607 607        Cs_Database_Access_Iftypes.Db_Memory_Address_Type;
608 608     Access_To_File_Pointer_Record : Access_To_File_Pointer_Record_Type;
609 609         Fpr_Attributes : Fpr_Attributes_Type;
610 610     begin -- Field_Enable
611 611         Fpr_Attributes := Database_Header.Fpr_Attributes;
612 612     File_Pointer_Start_Address :=
613 613        Start_Address + Bytes_Per_Word *
614 614                    (Database_Header.First_Fpr_Address +
615 615                     Cs_Database_Access_Iftypes.Db_Memory_Address_Type
616 616                    (File_Number) *
617 617                     Cs_Database_Access_Iftypes.Db_Memory_Address_Type
618 618                    (Fpr_Attributes.Fpr_Size));
619 619     Access_To_File_Pointer_Record :=
620 620        To_Access_To_File_Pointer_Record_Type (File_Pointer_Start_Address);
621 621     return Portable_Types_Pkg.Unsigned_32
622 622           (Access_To_File_Pointer_Record.First_Word.Field_Enable_Bits);
623 623     end Field_Enable;
624 624 --
625 625 --
626 626 --!
627 627
628 628     function Database_Size
629 629         return Cs_Database_Access_Iftypes.Database_Size_Type is
630 630 --!
631 631 --      DESCRIPTION:    This function returns the size of the database.
632 632 --
633 633 --      DATA_RIGHTS: Honeywell ATSD Proprietary
634 634 --      ANCHOR: CS_CODE_1764
635 635 --      SHARED_DATA:
636 636 --              Name                                                    Mode
637 637 --              ================================================================
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
638 638 --              Database_Header                                         In
639 639 --              An access type to the database header
640 640 --      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
641 641 --!
642 642     begin
643 643     return Database_Header.Database_Size;
644 644     end Database_Size;
645 645
646 646     function Database_Version
647 647         return Cs_Database_Access_Iftypes.Database_Version_Type is
648 648 --!
649 649 --      DESCRIPTION:    This function returns the version of the database.
650 650 --
651 651 --      DATA_RIGHTS: Honeywell ATSD Proprietary
652 652 --      ANCHOR: CS_CODE_1765
653 653 --      SHARED_DATA:
654 654 --              Name                                                    Mode
655 655 --              ================================================================
656 656 --              Database_Header                                           In
657 657 --              An access type to the database header
658 658 --      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
659 659 --!
660 660     begin -- Database_Version
661 661     return Database_Header.Version;
662 662     end Database_Version;
663 663
664 664     function Crc_Value return Portable_Types_Pkg.Unsigned_32 is
665 665 --!
666 666 --      DESCRIPTION:    This function returns the CRC of the database.
667 667 --      DATA_RIGHTS: Honeywell ATSD Proprietary
668 668 --      ANCHOR: CS_CODE_1766
669 669 --      SHARED_DATA:
670 670 --              Name                                                    Mode
671 671 --              ================================================================
672 672 --              Database_Header                                         In
673 673 --              An access type to the database header
674 674 --      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
675 675 --!
676 676     begin
677 677     return Database_Header.Database_Crc;
678 678     end Crc_Value;
679 679
680 680
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
   681    681     function Creation_Date_Year return Cs_Database_Access_Iftypes.Year_Type is
   682    682 --!
   683    683 --      DESCRIPTION: Return the creation date year of a database
   684    684 --
   685    685 --      DATA_RIGHTS: Honeywell ATSD Proprietary
   686    686 --      ANCHOR: CS_CODE_1767
   687    687 --      SHARED_DATA:
   688    688 --                Name                                                  Mode
   689    689 --                ====================================================================
   690    690 --                Database_Header                                       In
   691    691 --                An access type to the database header
   692    692 --      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
   693    693 --!
   694    694       Creation_Date : Creation_Date_Type;
   695    695     begin
   696    696         Creation_Date := Database_Header.Creation_Date;
   697    697     return Cs_Database_Access_Iftypes.Year_Type
   698    698                   (Creation_Date.Year);
   699    699     end Creation_Date_Year;
   700    700
   701    701     function Creation_Date_Month return Cs_Database_Access_Iftypes.Month_Type is
   702    702 --!
   703    703 --      DESCRIPTION: Return the creation date month of a database
   704    704 --
   705    705 --      DATA_RIGHTS: Honeywell ATSD Proprietary
   706    706 --      ANCHOR: CS_CODE_1768
   707    707 --      SHARED_DATA:
   708    708 --                Name                                                  Mode
   709    709 --                ====================================================================
   710    710 --                Database_Header                                       In
   711    711 --                An access type to the database header
   712    712 --      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
   713    713 --!
   714    714       Creation_Date : Creation_Date_Type;
   715    715     begin
   716    716         Creation_Date := Database_Header.Creation_Date;
   717    717     return Cs_Database_Access_Iftypes.Month_Type
   718    718           (Creation_Date.Month);
   719    719     end Creation_Date_Month;
   720    720
   721    721     function Creation_Date_Day return Cs_Database_Access_Iftypes.Day_Type is
   722    722 --!
   723    723 --      DESCRIPTION: Return the creation date day of a database
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
724 | 724 |--
725 | 725 |--      DATA_RIGHTS: Honeywell ATSD Proprietary
726 | 726 |--      ANCHOR: CS_CODE_1769
727 | 727 |--      SHARED_DATA:
728 | 728 |--              Name                                                    Mode
729 | 729 |--              ================================================================
730 | 730 |--              Database_Header                                         In
731 | 731 |--              An access type to the database header
732 | 732 |--      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
733 | 733 |--!
734 | 734 |      Creation_Date : Creation_Date_Type;
735 | 735 |    begin
736 | 736 |        Creation_Date := Database_Header.Creation_Date;
737 | 737 |    return Cs_Database_Access_Iftypes.Day_Type
738 | 738 |          (Creation_Date.Day);
739 | 739 |    end Creation_Date_Day;
740 | 740 |
741 | 741 |    function Database_Identifier
742 | 742 |        return Cs_Database_Access_Iftypes.Identifier_Type is
743 | 743 |--!
744 | 744 |--      DESCRIPTION: Return the identifer of a database
745 | 745 |--
746 | 746 |--      DATA_RIGHTS: Honeywell ATSD Proprietary
747 | 747 |--      ANCHOR: CS_CODE_1770
748 | 748 |--      SHARED_DATA:
749 | 749 |--               Name                                                   Mode
750 | 750 |--              ================================================================
751 | 751 |--              Database_Header                                         In
752 | 752 |--              An access type to the database header
753 | 753 |--      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_Header has been called previously.
754 | 754 |--!
755 | 755 |    begin
756 | 756 |    return Database_Header.Database_Identifier;
757 | 757 |    end Database_Identifier;
758 | 758 |
759 | 759 |    package body Cs_File_Access_Gnrc is
760 | 760 |--!
761 | 761 |--  PURPOSE:     This package contains types for use with routines for accessing information in databases which follow
    |     |»  the 777
762 | 762 |--              AIMS Common Database Format (reference Honeywell CM AIMS-H-01320).
763 | 763 |--  DATA_RIGHTS: Honeywell ATSD Proprietary
764 | 764 |--  ANCHOR:      CS_CODE_1771
765 | 765 |--  RAISES:      N/A
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
766 766 --  SPECIAL_CONSIDERATIONS:
767 767 --      This package is instantiated with a file number. It assumes procedure Initialize_Header in package CS_Database
        » _Access_GNRC
768 768 --      has been called previous to it's instantiation.
769 769 --!
770 770     File_Pointer_For_Record_Access_Start_Address :
771 771        Cs_Database_Access_Iftypes.Db_Memory_Address_Type;
772 772     Record_Size_For_Record_Access :
773 773        Cs_Database_Access_Iftypes.Record_Size_Type;
774 774     Pointer_To_File_For_Record_Access :
775 775        Cs_Database_Access_Iftypes.Db_Memory_Address_Type;
776 776     File_Size_Or_Record_Count_For_Record_Access :
777 777        Cs_Database_Access_Iftypes.File_Size_Type;
778 778     File_Exists_For_Record_Access : Boolean;
779 779 --          Objects to set up global (to this package) values for access type to file pointer record, record size, poi
        » nter to file,
780 780 --          file size/record count and file exists. This initialization is done in procedure Initialize_File.
781 781
782 782     procedure Initialize_File is
783 783 --!
784 784 --      DESCRIPTION:     This procedure will set up the record size, pointer to file, file size/record count and file
785 785 --                       exists for the file with which the package CS_File_Access_GNRC is instantiated.
786 786 --
787 787 --      DATA_RIGHTS: Honeywell ATSD Proprietary
788 788 --      ANCHOR: CS_CODE_1772
789 789 --      SHARED_DATA:
790 790 --                Name                                                    Mode
791 791 --               ================================================================
792 792 --               File_Num_For_Record_Access                              In
793 793 --                 File number which will be used for access- generic parameter to CS_File_Access_GNRC
794 794 --               CS_Database_Access_GNRC.Database_Header                 In
795 795 --                 An access type to the database header
796 796 --               Record_Size_For_Record_Access                           Out
797 797 --                 Record size from the file pointer record
798 798 --               Pointer_To_File_For_Record_Access                       Out
799 799 --                 An access type to the database file
800 800 --               File_Size_Or_Record_Count_For_Record_Access             Out
801 801 --                 File size or record count from the file pointer record
802 802 --               File_Exists_For_Record_Access                           Out
803 803 --                 File exists flag from file pointer record
804 804 --
805 805 --      SPECIAL_CONSIDERATIONS: N/A
806 806 --!
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
807 807         File_Pointer_Record : Access_To_File_Pointer_Record_Type;
808 808             Fpr_Attributes : Fpr_Attributes_Type;
809 809             First_Word     : First_Word_Type;
810 810
811 811     begin -- procedure Initialize_File
812 812             Fpr_Attributes := Cs_Database_Access_Gnrc.Database_Header.Fpr_Attributes;
813 813
814 814 -- Per CSW #598, this procedure will default all output data if asked to work on
815 815 -- on a file shose FPR isn't present.
816 816 --
817 817         if (Portable_Types_Pkg.Unsigned_16 (File_Num_For_Record_Access) <
818 818         Fpr_Attributes.Num_Of_Fprs) then
819 819         File_Pointer_For_Record_Access_Start_Address :=
820 820            Start_Address + Bytes_Per_Word *
821 821                       (Cs_Database_Access_Gnrc.
822 822                        Database_Header.First_Fpr_Address +
823 823                        Cs_Database_Access_Iftypes.
824 824                       Db_Memory_Address_Type
825 825                       (File_Num_For_Record_Access) *
826 826                        Db_Memory_Address_Type
827 827                       (Fpr_Attributes.Fpr_Size));
828 828         File_Pointer_Record :=
829 829            To_Access_To_File_Pointer_Record_Type
830 830               (File_Pointer_For_Record_Access_Start_Address);
831 831                 First_Word := File_Pointer_Record.First_Word;
832 832
833 833         -- Record Size for Record Access is stored as number of bytes to save multiplication for every access
834 834         Record_Size_For_Record_Access :=
835 835            Cs_Database_Access_Iftypes.Record_Size_Type
836 836               (First_Word.Record_Size) * Bytes_Per_Word;
837 837
838 838         File_Size_Or_Record_Count_For_Record_Access :=
839 839            File_Pointer_Record.File_Size_Or_Rec_Count;
840 840         File_Exists_For_Record_Access :=
841 841            File_Pointer_Record.First_Word.File_Exists;
842 842
843 843         Pointer_To_File_For_Record_Access :=
844 844            Start_Address + Bytes_Per_Word *
845 845                       File_Pointer_Record.Pointer_To_File;
846 846         else
847 847         -- FPR doesn't exist for the requested file, so default all output data to zero.
848 848         Record_Size_For_Record_Access := 0;
849 849         File_Size_Or_Record_Count_For_Record_Access := 0;
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
850 850         Pointer_To_File_For_Record_Access := 0;
851 851         File_Exists_For_Record_Access := False;
852 852         File_Pointer_For_Record_Access_Start_Address := 0;
853 853         end if; -- File_Num <= Num_Of_Fprs
854 854
855 855     end Initialize_File;
856 856
857 857
858 858     function Are_Extra_Words_Zero
859 859             (Record_Offset : in
860 860             Cs_Database_Access_Iftypes.Db_Memory_Address_Type;
861 861              Expected_Size_In_Words : in
862 862             Cs_Database_Access_Iftypes.Record_Size_Type)
863 863             return Boolean is
864 864 --!
865 865 -- ANCHOR: CS_CODE_1773
866 866 -- PURPOSE: See spec
867 867 -- REVISION_HISTORY:
868 868 -- Date             SCR #         Programmer
869 869 -- 6/30/94          399            Kevin Tucker
870 870 -- Initial development
871 871 -- 8/23/94          399            Kevin Tucker
872 872 -- Removed -1 from assignment to File_Size_In_Words.
873 873 -- 8/24/94          399            Kevin Tucker
874 874 -- Changed to verify that the extra words in a particular *record* are zero
875 875 -- rather than in a complete file.  This is so that the file can be expanded
876 876 -- to hold additional records and still pass the compatibility check.
877 877 --
878 878         Check_Passed : Boolean;
879 879         Record_Size_In_Words : Cs_Database_Access_Iftypes.Record_Size_Type;
880 880
881 881         --| @UNITS 32-bit words.
882 882         Delta_Size : Cs_Database_Access_Iftypes.Record_Size_Type;
883 883
884 884         Current_Ptr : Cs_Database_Access_Iftypes.Db_Memory_Address_Type;
885 885         Data_Word : Portable_Types_Pkg.Unsigned_32;
886 886         type Access_Unsigned_32 is access Portable_Types_Pkg.Unsigned_32;
887 887         Data_Word_Addr : Access_Unsigned_32;
888 888
889 889         function To_Access_Unsigned_32 is
890 890            new Unchecked_Conversion (Source => Cs_Database_Access_Iftypes.
891 891                                Db_Memory_Address_Type,
892 892                      Target => Access_Unsigned_32);
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
893 893     begin
894 894         Check_Passed :=
895 895            True; -- Assume the best (I'm an optimist,. and it also
896 896              -- makes the checking logic easier)
897 897
898 898         Current_Ptr := Pointer_To_File_For_Record_Access +
899 899                   (Record_Offset * Bytes_Per_Word);
900 900         -- Compute pointer to the first word of the record (the length word)
901 901         --
902 902         Data_Word_Addr := To_Access_Unsigned_32 (Current_Ptr);
903 903         Record_Size_In_Words := Data_Word_Addr.all;
904 904         -- Retrieve the record length word from memory.
905 905
906 906         Delta_Size := Record_Size_In_Words - Expected_Size_In_Words;
907 907         Current_Ptr := Current_Ptr +
908 908                   (Expected_Size_In_Words * Bytes_Per_Word);
909 909
910 910         -- Loop "Check_For_Non_Zero_Words" satifies requirement CS_DR_0632
911 911         Check_For_Non_Zero_Words:
912 912         while Delta_Size > 0 loop
913 913             Data_Word_Addr := To_Access_Unsigned_32 (Current_Ptr);
914 914             Data_Word := Data_Word_Addr.all;
915 915             Check_Passed := Check_Passed and then (Data_Word = 0);
916 916             exit when (not Check_Passed);
917 917
918 918             Delta_Size := Delta_Size - 1;
919 919
920 920             Current_Ptr := Current_Ptr + Bytes_Per_Word;
921 921
922 922         end loop Check_For_Non_Zero_Words;
923 923         return Check_Passed;
924 924
925 925     end Are_Extra_Words_Zero;
926 926
927 927     procedure Access_Fixed_Length_By_Record_Number_Gnrc
928 928              (Record_Number : in
929 929              Cs_Database_Access_Iftypes.Record_Num_Type;
930 930               Access_Pointer : out Access_To_Record_Type) is
931 931 --!
932 932 --      DESCRIPTION:     This generic procedure will allow retrieval of a record from a file of fixed length records g
        » iven a
933 933 --                       record number. An access type to the desired record will be returned. A record type and an
934 934 --                       access type to the record type to be retrieved is defined at instantiation. It is designed to
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
» be
935 935 --                      instantiated for every record type which will be retrieved from the database file specified a
» t
936 936 --                      instantiation of this package.
937 937 --
938 938 --      DATA_RIGHTS: Honeywell ATSD Proprietary
939 939 --      ANCHOR: CS_CODE_1774
940 940 --      SHARED_DATA:
941 941 --               Name                                                  Mode
942 942 --              ============================================================
943 943 --              Pointer_To_File_For_Record_Access                     In
944 944 --              An access type to the database file
945 945 --              Record_Size_For_Record_Access                         In
946 946 --                Record size from the file pointer record
947 947 --      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_File has been called previously.
948 948 --!
949 949
950 950         function To_Access_To_Record_Type is
951 951            new Unchecked_Conversion (Source => Db_Memory_Address_Type,
952 952                        Target => Access_To_Record_Type);
953 953         Record_Pointer : Db_Memory_Address_Type;
954 954
955 955     begin
956 956         Record_Pointer := Pointer_To_File_For_Record_Access +
957 957                   Record_Size_For_Record_Access * Record_Number;
958 958         Access_Pointer := To_Access_To_Record_Type (Record_Pointer);
959 959     end Access_Fixed_Length_By_Record_Number_Gnrc;
960 960
961 961     procedure Access_Next_Fixed_Record_Gnrc
962 962               (Access_Pointer : in out Access_To_Record_Type) is
963 963 --!
964 964 --      DESCRIPTION:     This generic procedure will allow retrieval of the next record from a file of fixed length re
» cords
965 965 --                      given an access type to the previous record.  An access type to the desired record will be re
» turned.
966 966 --                      A record type and an access type to the record type to be retrieved is defined at instantiati
» on.
967 967 --                      It is designed to be instantiated for every record type which will be retrieved from the data
» base
968 968 --                      file specified at instantiation of this package.
969 969 --
970 970 --      DATA_RIGHTS: Honeywell ATSD Proprietary
971 971 --      ANCHOR: CS_CODE_1775
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
   972|    972|--      SHARED_DATA:
   973|    973|--                Name                                                  Mode
   974|    974|--               ================================================================
   975|    975|--               Pointer_To_File_For_Record_Access                     In
   976|    976|--               An access type to the database file
   977|    977|--               Record_Size_For_Record_Access                         In
   978|    978|--                 Record size from the file pointer record
   979|    979|--      SPECIAL_CONSIDERATIONS: This procedure assumes procedure Initialize_File has been called previously.
   980|    980|--!
   981|    981|
   982|    982|        function To_Access_To_Record_Type is
   983|    983|           new Unchecked_Conversion (Source => Db_Memory_Address_Type,
   984|    984|                     Target => Access_To_Record_Type);
   985|    985|        function To_Memory_Address_Type is
   986|    986|           new Unchecked_Conversion (Source => Access_To_Record_Type,
   987|    987|                     Target => Db_Memory_Address_Type);
   988|    988|        Record_Pointer : Db_Memory_Address_Type;
   989|    989|
   990|    990|    begin
   991|    991|        Record_Pointer := To_Memory_Address_Type (Access_Pointer) +
   992|    992|                  Record_Size_For_Record_Access;
   993|    993|        Access_Pointer := To_Access_To_Record_Type (Record_Pointer);
   994|    994|    end Access_Next_Fixed_Record_Gnrc;
   995|    995|
   996|    996|    procedure Access_Variable_Length_By_Record_Number_Gnrc
   997|    997|              (Record_Number : in
   998|    998|              Cs_Database_Access_Iftypes.Record_Num_Type;
   999|    999|               Access_Pointer : out Access_To_Record_Type) is
  1000|   1000|--!
  1001|   1001|--      DESCRIPTION:     This generic procedure will allow retrieval of a record from a file of variable length record
        |       |» s given a
  1002|   1002|--                       record number. An access type to the desired record will be returned. A record type and
  1003|   1003|--                       access type to the record type to be retrieved is defined at instantiation. It is designed to
        |       |»  be
  1004|   1004|--                       instantiated for every record type which will be retrieved from the database file specified a
        |       |» t
  1005|   1005|--                       instantiation of this package.
  1006|   1006|--
  1007|   1007|--      DATA_RIGHTS: Honeywell ATSD Proprietary
  1008|   1008|--      ANCHOR: CS_CODE_1776
  1009|   1009|--      SHARED_DATA:
  1010|   1010|--                Name                                                  Mode
  1011|   1011|--               ================================================================
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
1012 1012 --              Pointer_To_File_For_Record_Access                  In
1013 1013 --              An access type to the database file
1014 1014 --      SPECIAL_CONSIDERATIONS: N/A
1015 1015 --!
1016 1016         type Access_Db_Memory_Address_Type is
1017 1017            access Cs_Database_Access_Iftypes.Db_Memory_Address_Type;
1018 1018
1019 1019         function To_Access_To_Record_Type is
1020 1020            new Unchecked_Conversion (Source => Db_Memory_Address_Type,
1021 1021                      Target => Access_To_Record_Type);
1022 1022         function To_Access_Db_Memory_Address_Type is
1023 1023            new Unchecked_Conversion
1024 1024               (Source => Db_Memory_Address_Type,
1025 1025                Target => Access_Db_Memory_Address_Type);
1026 1026         Record_Pointer : Db_Memory_Address_Type;
1027 1027         Access_Variable_Length_Size : Access_Db_Memory_Address_Type;
1028 1028
1029 1029     begin
1030 1030         Record_Pointer :=
1031 1031            Pointer_To_File_For_Record_Access;    -- set pointer to first variable length record
1032 1032         for Record_Num in
1033 1033            1 .. (Record_Number) loop             -- loop through number of records
1034 1034         Access_Variable_Length_Size :=
1035 1035            To_Access_Db_Memory_Address_Type (Record_Pointer);
1036 1036         Record_Pointer :=
1037 1037            Record_Pointer +
1038 1038               Bytes_Per_Word *
1039 1039              Access_Variable_Length_Size.
1040 1040                 all;   -- add the size of the variable length data
1041 1041         end loop;
1042 1042         Access_Pointer := To_Access_To_Record_Type (Record_Pointer);
1043 1043     end Access_Variable_Length_By_Record_Number_Gnrc;
1044 1044
1045 1045     procedure Access_By_Offset_Gnrc
1046 1046               (Record_Offset : in Cs_Database_Access_Iftypes.
1047 1047                         Db_Memory_Address_Type;
1048 1048                Access_Pointer : out Access_To_Record_Type) is
1049 1049 --!
1050 1050 --      DESCRIPTION:     This generic procedure will allow retrieval of a record given an offset.
1051 1051 --                       An access type to the desired record wil be returned. A record type and an access type to the
      »  desired
1052 1052 --                       record is defined at instantiation. It is designed to be instantiated for every record type w
      » hich will be
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
1053 1053 --                       retrieved from the database file specified at instantiation of this package.
1054 1054 --
1055 1055 --      DATA_RIGHTS: Honeywell ATSD Proprietary
1056 1056 --      ANCHOR: CS_CODE_1777
1057 1057 --      SHARED_DATA:
1058 1058 --                Name                                                  Mode
1059 1059 --               ================================================================
1060 1060 --               Pointer_To_File_For_Record_Access                     In
1061 1061 --               An access type to the database file
1062 1062 --      SPECIAL_CONSIDERATIONS: N/A
1063 1063 --!
1064 1064
1065 1065         function To_Access_To_Record_Type is
1066 1066            new Unchecked_Conversion (Source => Db_Memory_Address_Type,
1067 1067                      Target => Access_To_Record_Type);
1068 1068         Record_Pointer : Cs_Database_Access_Iftypes.Db_Memory_Address_Type;
1069 1069
1070 1070     begin
1071 1071         Record_Pointer := Pointer_To_File_For_Record_Access +
1072 1072                   Bytes_Per_Word * Record_Offset;
1073 1073         Access_Pointer := To_Access_To_Record_Type (Record_Pointer);
1074 1074     end Access_By_Offset_Gnrc;
1075 1075
1076 1076     procedure Access_By_Byte_Offset_Gnrc
1077 1077               (Record_Offset : in Cs_Database_Access_Iftypes.
1078 1078                           Db_Memory_Address_Type;
1079 1079                Access_Pointer : out Access_To_Record_Type) is
1080 1080 --!
1081 1081 --      DESCRIPTION:     This generic procedure will allow retrieval of a record given a byte offset.
1082 1082 --                       An access type to the desired record wil be returned. A record type and an access type to the
      »  desired
1083 1083 --                       record is defined at instantiation. It is designed to be instantiated for every record type w
      » hich will be
1084 1084 --                       retrieved from the database file specified at instantiation of this package.
1085 1085 --
1086 1086 --      DATA_RIGHTS: Honeywell ATSD Proprietary
1087 1087 --      ANCHOR: CS_CODE_1778
1088 1088 --      SHARED_DATA:
1089 1089 --                Name                                                  Mode
1090 1090 --               ================================================================
1091 1091 --               Pointer_To_File_For_Record_Access                     In
1092 1092 --               An access type to the database file
1093 1093 --      SPECIAL_CONSIDERATIONS: N/A
```

File: CTP_A340S1A_PERF_CS_DB_ACCES_GNRC.STB (continued)

```
1094  1094 --!
1095  1095
1096  1096         function To_Access_To_Record_Type is
1097  1097            new Unchecked_Conversion (Source => Db_Memory_Address_Type,
1098  1098                      Target => Access_To_Record_Type);
1099  1099         Record_Pointer : Cs_Database_Access_Iftypes.Db_Memory_Address_Type;
1100  1100
1101  1101     begin
1102  1102         Record_Pointer := Pointer_To_File_For_Record_Access + Record_Offset;
1103  1103         Access_Pointer := To_Access_To_Record_Type (Record_Pointer);
1104  1104     end Access_By_Byte_Offset_Gnrc;
1105  1105
1106  1106     end Cs_File_Access_Gnrc;
1107  1107 end Cs_Database_Access_Gnrc;
```

Mode: All Lines

File: CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TRT

```
!*****************************************************************************************************************
!*
!* TRACE FILENAME        :  CTP_A340S1A_PERF_DESPATH_CALC_CC_RATE.TRT
!*
!* MODIFICATION HISTORY  :
!*                           DATE           SCR #          AUTHOR              DESCRIPTION
!*                           =====          =====          ======              ===========
!*
!*                        Aug 12, 2010     52527.07     Yanfei Shen     Initial Development for A340 S1A S1 plan.
!*                                                                      1. Rollover from A320
!*                                                                         CTP_A320_PERF_DESPATH_CALC_CC_RATE.TRT;4.
!*****************************************************************************************************************
A340    SRD    A340_PERF_TEST_2351   PERF_SRD_9751
A340    SRD    A340_PERF_TEST_2351   PERF_SRD_9752
A340    SRD    A340_PERF_TEST_2351   PERF_SRD_2721
A340    SRD    A340_PERF_TEST_2351   PERF_SRD_2043
A340    SRD    A340_PERF_TEST_2351   PERF_SRD_7473
A340    SRD    A340_PERF_TEST_2351   PERF_SRD_9646
A340    SDD    A340_PERF_TEST_2351   PERF_SDD_1576
A340    SDD    A340_PERF_TEST_2351   PERF_SDD_1577
A340    SDD    A340_PERF_TEST_2351   PERF_SDD_1578
A340    SDD    A340_PERF_TEST_2351   PERF_SDD_1579_INT
A340    SDD    A340_PERF_TEST_2351   PERF_SDD_1580
A340    SDD    A340_PERF_TEST_2351   PERF_SDD_1581_INT
A340    SDD    A340_PERF_TEST_2351   PERF_SDD_3571
```