ECMPS Emissions Check Specifications

United States Environmental Protection Agency
Office of Air and Radiation
Clean Air Markets Division
Ariel Rios Building
1200 Pennsylvania Avenue
Washington, DC 20460

December 02, 2020

Table of Contents

Appendix D and E Status	1
Daily Calibration Status	37
Daily Calibration Test	50
Daily Emissions Data	104
Daily Interference Status	120
Daily Test	127
EM Weekly System Integrity Test	134
EM Weekly Test Summary	146
Emissions Audit Checks	156
Flow-to-Load Status	158
Hourly Aggregation	172

Hourly Appendix D 219

Hourly Appendix E 282

Hourly Apportionment 299

Hourly Calculated Data 326

Hourly Derived Data 387

Hourly General 455

Hourly Monitor Data 499

Hourly Operating Data 577

Leak Status 655

Linearity Status 664

LME 698

MATS Calculated Hourly Value Checks 752

MATS Derived Hourly Value Checks 780

MATS Hourly GFM Data 802

MATS Monitor Hourly Value Checks 812

MATS Operating Hour Checks 847

MATS Sampling Train Checks 873

MATS Sorbent Trap Data 897

NSPS4T Summary, Compliance Period and 917

Annual Checks 924

RATA Status 968

Weekly System Integrity Status

Check Category:

Appendix D and E Status

Check Code: ADESTAT-1

Check Name: Determine Appendix E Status

Related Former Checks:

Applicability:

Description:

Validation Tables:

Fuel Code (Lookup Table)

Specifications:

Set ***PriorAppendixERecord*** = null.
Set ***InvalidAppendixERecord*** = null.
Set ***CurrentAppendixEStatus*** = null.
Set ***PriorAppendixEEventRecord*** = null.
Set ***SubsequentAppendixERecord*** = null
Set ***AppendixEMissingOpDataInfo*** = null.

If (***App E Op Code*** in set {N, W, X, Y, Z}) AND ***AppE NOXE System ID*** is not null)

Append ***AppE NOXE System ID*** to ***NOXE System ID Array.***

Locate the most recent record in ***AppendixETestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***AppE NOXE System ID*** and the TestResultCode is not equal to "INVALID" and the EndDate/Hour is prior to the ***CurrentOperatingDate/Hour***.

if (***AppendixETestRecordsByLocationForQAStatus*** is found)

Set ***PriorAppendixERecord*** = the found record in ***AppendixETestRecordsByLocationForQAStatus***.

Locate the most recent record in ***AppendixETestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***AppENOXESystemID*** and the TestResultCode is equal to "INVALID" and the EndDate/Hour is prior to the ***CurrentOperatingDate/Hour*** and the EndDate/Hour is greater than the ***PriorAppendixERecord***.EndDate/Hour.

if (***AppendixETestRecordsByLocationForQAStatus*** is found)

Set ***InvalidAppendixERecord*** = the found record in ***AppendixETestRecordsByLocationForQAStatus***.

else

Locate the most recent record in ***AppendixETestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***AppE NOXE System ID*** and the TestResultCode is equal to "INVALID" and the EndDate/Hour is prior to the ***CurrentOperatingDate/Hour***.

if (***AppendixETestRecordsByLocationForQAStatus*** is found)

Set ***InvalidAppendixERecord*** = the found record in ***AppendixETestRecordsByLocationForQAStatus***.

if (***PriorAppendixERecord*** is not null)

if (***PriorAppendixERecord***.QANeedsEvaluationFlag = "Y")

Set ***CurrentAppendixEStatus*** = "Prior Test Not Yet Evaluated".

else if (***PriorAppendixERecord***.TestResultCode = null)

Set ***CurrentAppendixEStatus*** = "OOC-Prior Test Has Critical Errors".

else

Set *PriorTestExpirationDate* = ***PriorAppendixERecord***.TestExpirationDate

if (*PriorTestExpirationDate* is null)

Set *PriorTestExpirationDate* = 5 years (20 calendar quarters) after the end of the quarter of the ***PriorAppendixERecord***.EndDate/Hour.

if (***CurrentOperatingDate/Hour*** is AFTER the *PriorTestExpirationDate*)

Set ***CurrentAppendixEStatus*** = "OOC-Expired".

else

Locate the most recent record in ***QACertificationEventRecords*** where the SystemID is equal to the ***AppENOXESystemID*** and the RequiredTestCode is equal to 75 and the QACertEventDate/Hour is prior to the ***CurrentOperatingDate/Hour*** and theQACertEventDate/Hour is on or after the ***PriorAppendixERecord***.EndDate/Hour

if (***QACertificationEventRecords*** is found)

Set ***PriorAppendixEEventRecord*** = the found record in ***QACertificationEventRecords***.

if (the number of calendar days ON OR AFTER the ***PriorAppendixEEventRecord***.QACertEventDate and ON OR BEFORE the ***CurrentOperatingDate/Hour*** > 180)

Set ***CurrentAppendixEStatus*** = "OOC-Event".

else if (***PriorAppendixEEventRecord*** .MinOpDaysPriorQuarter is null)
Set ***PriorAppendixEEventRecord*** .MinOpDaysPriorQuarter = 0
Set ***PriorAppendixEEventRecord*** .MaxOpDaysPriorQuarter = 0

for each quarter beginning with the quarter of the ***PriorAppendixEEventRecord***.QACertEventDate and continuing through the quarter BEFORE the ***CurrentOperatingDate/Hour***:

if (***EarliestLocationReportDate*** <= the last day of the quarter being checked)

Locate the record in ***OperatingSuppDataRecordsbyLocation*** where the OpTypeCode is equal to "OPDAYS" and the reporting period is equal to the quarter being checked.

if (***OperatingSuppDataRecordsbyLocation*** is not found)

Set ***PriorAppendixEEventRecord***.MinOpDaysPriorQuarter = -1
Set ***AppendixEMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.
exit for.

else

If the quarter being checked is the quarter of the

PriorAppendixEEventRecord.QACertEventDate

If (***OperatingSuppDataRecordsbyLocation***.OpValue MINUS the number of calendar days in the quarter being checked that are PRIOR to the ***PriorAppendixEEventRecord***.QACertEventDate > 0)

Set ***PriorAppendixEEventRecord***.MinOpDaysPriorQuarter = ***OperatingSuppDataRecordsbyLocation***.OpValue MINUS the number of calendar days in the quarter being checked that are PRIOR to the ***PriorAppendixEEventRecord***.QACertEventDate

If (***OperatingSuppDataRecordsbyLocation***.OpValue is less than the number of calendar days in the quarter being checked that are ON OR AFTER the ***PriorAppendixEEventRecord***.QACertEventDate)

Set ***PriorAppendixEEventRecord***.MaxOpDaysPriorQuarter = ***OperatingSuppDataRecordsbyLocation***.OpValue.

else

Set ***PriorAppendixEEventRecord***.MaxOpDaysPriorQuarter = the number of calendar days in the quarter being checked that are ON OR AFTER the ***PriorAppendixEEventRecord***.QACertEventDate.

else

Set ***PriorAppendixEEventRecord***.MinOpDaysPriorQuarter = ***PriorAppendixEEventRecord***.MinOpDaysPriorQuarter + ***OperatingSuppDataRecordsbyLocation***.OpValue.

Set ***PriorAppendixEEventRecord***.MaxOpDaysPriorQuarter = ***PriorAppendixEEventRecord***.MaxOpDaysPriorQuarter + ***OperatingSuppDataRecordsbyLocation***.OpValue.

If (***CurrentAppendixEStatus*** does NOT begin with "OOC")

if (***Rpt Period Op Time Accumulator Array*** for the Location == -1)

set ***CurrentAppendixEStatus*** = "Invalid Op Data"

elseif (***PriorAppendixEEventRecord***.MinOpDaysPriorQuarter == -1)

set ***CurrentAppendixEStatus*** = "Missing Op Data"

else if (***PriorAppendixEEventRecord***.MinOpDaysPriorQuarter + ***Rpt Period Op Days Accumulator Array*** for the Location > 30)

Set ***CurrentAppendixEStatus*** = "OOC-Event".

else if (***PriorAppendixEEventRecord***.MaxOpDaysPriorQuarter + ***Rpt Period Op Days Accumulator Array*** for the Location > 30)

Set ***CurrentAppendixEStatus*** = "Undetermined-Event".

else

Set ***CurrentAppendixEStatus*** = "IC".

else

Set ***CurrentAppendixEStatus*** = "IC".

else

if (***AppEFuelCode*** is not equal to "MIX")

Locate the earliest record in ***AppendixETestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***AppENOXESystemID*** and the TestResultCode is not equal to "INVALID" and the EndDate/Hour is on or after the ***CurrentOperatingDate/Hour***

if (***AppendixETestRecordsByLocationForQAStatus*** is found)

Set ***SubsequentAppendixERecord*** = the found record in ***AppendixETestRecordsByLocationForQAStatus***.

Locate the earliest record in ***AppendixETestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***AppENOXESystemID*** and the TestResultCode is equal to "INVALID" and the EndDate/Hour is on or after the ***CurrentOperatingDate/Hour*** and the EndDate/Hour is before the ***SubsequentAppendixERecord***.EndDate/Hour

if (***AppendixETestRecordsByLocationForQAStatus*** is found)

Set ***InvalidAppendixERecord*** = the found record in ***AppendixETestRecordsByLocationForQAStatus***.

Locate the earliest record in ***OperatingSuppDataRecordsbyLocation*** where the FuelCode is equal to ***AppEFuelCode*** and the OpTypeCode is equal to "OPHOURS"

If ***OperatingSuppDataRecordsbyLocation*** is found)

Set *FuelOpSuppDataRecord* = the found record in ***OperatingSuppDataRecordsbyLocation***
Set *DateFuelFirstCombusted* = end date of quarter of the *FuelOpSuppDataRecord* .RptPeriodID - int((*FuelOpSuppDataRecord* .OpValue - 1)/24) days

if (If ***OperatingSuppDataRecordsbyLocation*** is found AND *DateFuelFirstCombusted* is more than 180 calendar days before the ***CurrentOperatingDate/Hour***)

Set ***CurrentAppendixEStatus*** = "OOC-No Prior Test"

else

Locate the record in the ***FuelCode*** lookup table where the FuelCode is equal to ***AppEFuelCode.***

Locate the record in ***FuelRecordsByHourLocation*** for the hour and location where the FuelCode is equal to the ***FuelCode.***UnitFuelCode

if (***FuelRecordsByHourLocation*** is not found OR more than one ***FuelRecordsByHourLocation*** is found)

Set ***CurrentAppendixEStatus*** = "Invalid Location Fuel"

else if (***FuelRecordsByHourLocation***.IndicatorCode is equal to "S",

if (***SubsequentAppendixERecord***.QANeedsEvaluationFlag = "Y")

Set ***CurrentAppendixEStatus*** = "Subsequent Test Not Yet Evaluated".

else if (***SubsequentAppendixERecord***.TestResultCode = null)

Set ***CurrentAppendixEStatus*** = "OOC-Subsequent Test Has Critical Errors".

else

Set ***CurrentAppendixEStatus*** = "IC"
Set ***PriorAppendixERecord*** = ***SubsequentAppendixERecord***

else

Set ***CurrentAppendixEStatus*** = "OOC-No Prior Test"

else

Set ***CurrentAppendixEStatus*** = "OOC-No Prior Test"

Locate the earliest record in ***AppendixETestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***AppENOXESystemID*** and the TestResultCode is equal to "INVALID" and the EndDate/Hour is on or after the ***CurrentOperatingDate/Hour***

if (***AppendixETestRecordsByLocationForQAStatus*** is found)

Set ***InvalidAppendixERecord*** = the found record in ***AppendixETestRecordsByLocationForQAStatus***.

else

Set ***CurrentAppendixEStatus*** = "OOC-No Prior Test"

if (***CurrentAppendixEStatus*** begins with "OOC")

if (***InvalidAppendixERecord*** is not null)

Set ***CurrentAppendixEStatus*** = ***CurrentAppendixEStatus*** & "*".

if (***CurrentAppendixEStatus*** does not begin with "IC")
Return result ***CurrentAppendixEStatus***.

Results:

Result	Response	Severity
Invalid Location Fuel	The Appendix E test status for MonitoringSystemID [ID] could not be determined, because you did not report a single, valid unit fuel record for FuelCode [unitfuel] that was active during the current hour.	Critical Error Level 1
Invalid Monitor System	The Appendix E test status for MonitoringSystemID [ID] could not be determined, because the Monitor System record for the NOXE system has a critical error.	Critical Error Level 1
Invalid Op Data	The Appendix E test status for MonitoringSystemID [ID] could not be determined, because the OperatingTime in at least one Hourly Operating Data records was missing or invalid.	Critical Error Level 1
Missing Fuel Op Data	The Appendix E test status for MonitoringSystemID [ID] could not be determined, because the Op Supp Data record for OPHOURS for FuelCode [fuel] is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
Missing Op Data	The Appendix E test status for MonitoringSystemID [ID] could not be determined, because the Op Supp Data record for OPDAYS is missing for [MISSINGOPDATAINFO] (and possibly other previous reporting periods). If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
OOC-Event	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for MonitoringSystemID [ID], but you did not perform a subsequent Appendix E test within the specified timeframe.	Critical Error Level 1
OOC-Event*	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for MonitoringSystemID [ID], but you did not perform a subsequent Appendix E test within the specified timeframe. An invalid Appendix E test was ignored.	Critical Error Level 1
OOC-Expired	The prior Appendix E test TestNumber [testnum] for MonitoringSystemID [ID] has expired.	Critical Error Level 1
OOC-Expired*	The prior Appendix E test TestNumber [testnum] for MonitoringSystemID [ID] has expired. An invalid prior Appendix E test TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-No Prior Test	You did not report a prior Appendix E test for MonitoringSystemID [ID].	Critical Error Level 1
OOC-No Prior Test*	You did not report a prior Appendix E test for MonitoringSystemID [ID]. An invalid prior Appendix E test Test Number [invtestnum] was ignored.	Critical Error Level 1
OOC-Prior Test Has Critical Errors	The applicable prior Appendix E test TestNumber [testnum] for MonitoringSystemID [ID] has critical errors.	Critical Error Level 1
OOC-Prior Test Has Critical Errors*	The prior Appendix E test TestNumber [testnum] for MonitoringSystemID [ID] has critical errors. An invalid prior Appendix E test TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Subsequent Test Has Critical Errors	The subsequent recertification Appendix E test TestNumber [subtestnum] for MonitoringSystemID [ID] has critical errors.	Critical Error Level 1
OOC-Subsequent Test Has Critical Errors*	The subsequent recertification Appendix E test TestNumber [subtestnum] for MonitoringSystemID [ID] has critical errors. An invalid Appendix E test TestNumber [invtestnum] was ignored.	Critical Error Level 1
Prior Test Not Yet Evaluated	The Appendix E test status for MonitoringSystemID [ID] could not be determined, because the applicable prior Appendix E test TestNumber [testnum] for the system has not yet been evaluated.	Critical Error Level 1
Subsequent Test Not Yet Evaluated	The Appendix E test status for MonitoringSystemID [ID] could not be determined, because the subsequent certification test TestNumber [subtestnum] for the system has not yet been evaluated.	Critical Error Level 1
Undetermined-Event	The software could not determine if the current hour was within the 30-operating day window required to conduct another Appendix E test following QACertEventCode [code] QACertEventDate [eventdate] for MonitoringSystemID [ID].	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation
	Conditions:	App E Checks Needed Equals true
2	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification
	Conditions:	App E Constant Fuel Mix Equals true
3	Process/Category:	Emissions Data Evaluation Report --------- Hourly Fuel Flow
	Conditions:	App E Constant Fuel Mix Equals false

Check Code: ADESTAT-2

Check Name: Locate Most Recent Prior Accuracy Test

Related Former Checks:

Applicability:

Description: Determines if there is an applicable prior accuracy test.

Specifications:

Set ***CurrentAccuracyStatus*** = null.
Set ***PriorAccuracyRecord*** = null.
Set ***InvalidAccuracyRecord*** = null.
Set ***InappropriateTransmitterTransducerTest*** to false.

Locate the most recent record in ***AccuracyTestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecordToCheck***.ComponentID and the TestResult is not equal to "INVALID" and the EndDate/Hour is prior to the ***CurrentOperatingDate/Hour***

if (***AccuracyTestRecordsByLocationForQAStatus*** is found)

Set ***PriorAccuracyRecord*** = the found record in ***AccuracyTestRecordsByLocationForQAStatus*** .

if (***PriorAccuracyRecord***.TestTypeCode is equal to "FFACCTT" AND ***FuelFlowComponentRecordToCheck***.SampleAcquisitionMethod is NOT equal to "ORF", "NOZ", or "VEN")

Set ***InappropriateTransmitterTransducerTest*** to true.

Locate the most recent record in ***AccuracyTestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecordToCheck***.ComponentID and the EndDate/Hour is prior to the ***CurrentOperatingDate/Hour*** and the EndDate/Hour is greater than the ***PriorAccuracyRecord***.EndDate/Hour and the TestResult is equal to "INVALID".

if (***AccuracyTestRecordsByLocationForQAStatus*** is found)

Set ***InvalidAccuracyRecord*** = the found record in ***AccuracyTestRecordsByLocationForQAStatus*** .

if (***PriorAccuracyRecord***.QANeedsEvaluationFlag = "Y")

Set ***CurrentAccuracyStatus*** = "Accuracy Test Not Yet Evaluated".

else if (***PriorAccuracyRecord***.TestResultCode is null)

Set ***CurrentAccuracyStatus*** = "OOC-Accuracy Test Has Critical Errors".

else if (***PriorAccuracyRecord***.TestResultCode = "FAILED")

Set ***CurrentAccuracyStatus*** = "OOC-Accuracy Test Failed".

else if (***PriorAccuracyRecord***.TestResultCode = "ABORTED")

Set ***CurrentAccuracyStatus*** = "OOC-Accuracy Test Aborted".

else

Set ***CurrentAccuracyStatus*** = "OOC-No Prior Accuracy Test"

Locate the most recent record in ***AccuracyTestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecordToCheck***.ComponentID and the TestResult is equal to "INVALID" and the EndDate/Hour is prior to the ***CurrentOperatingDate/Hour***.

if (***AccuracyTestRecordsByLocationForQAStatus*** is found)

Set ***InvalidAccuracyRecord*** = the found record in ***AccuracyTestRecordsByLocationForQAStatus***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Code: ADESTAT-3

Check Name: Locate Most Recent Prior Accuracy Event

Related Former Checks:

Applicability:

Description: Determines if there is a applicable prior event requiring an Accuracy test.

Specifications:

Set ***PriorAccuracyEventRecord*** = null.

If (***CurrentAccuracyStatus*** is null)

Locate the most recent record in ***QACertificationEventRecords*** WHERE

a) the ComponentID is equal to the ***FuelFlowComponentRecordToCheck***.ComponentID AND
b) FFACCRequired is equal to "Y" AND
c) the QACertEventDate/Hour is prior to the ***CurrentOperatingDate/Hour*** AND
d) the QACertEventDate/Hour is after the later of the ***PriorAccuracyRecord***.EndDate/Hour and the ***PriorAccuracyRecord***.ReinstallationDate/Hour.

if (***QACertificationEventRecords*** is found)

Set ***PriorAccuracyEventRecord*** = the found record in ***QACertificationEventRecords***.
Set ***CurrentAccuracyStatus*** = "OOC-Event".

else

Set *PriorTestExpirationDate* = 5 years (20 calendar quarters) after the end of the quarter of the ***PriorAccuracyRecord***.EndDate/Hour.

if (***CurrentOperatingDate/Hour*** is AFTER the *PriorTestExpirationDate*)

Set ***CurrentAccuracyStatus*** = "OOC-Accuracy Test Expired".

else

Set *PriorTestExpirationDate* = 4 quarters after the end of the quarter of the later of the ***PriorAccuracyRecord***.EndDate/Hour and the ***PriorAccuracyRecord***.ReinstallationDate/Hour.
Set ***PriorAccuracyRecord***.TestExpirationDate = *PriorTestExpirationDate.*

If (***CurrentOperatingDate/Hour*** is ON OR BEFORE the *PriorTestExpirationDate*)

Set ***CurrentAccuracyStatus*** = "IC".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Code: ADESTAT-4

Check Name: Determine Eligibility for Fuel Flow to Load Testing (Accuracy)

Related Former Checks:

Applicability:

Description: Determines if this component is eligible to extend their accuracy text expiration date using fuel flow to load testing.

Specifications:

Set ***FF2LAccuracyEligible*** = null.
Set ***FF2LAccuracyCheckDate*** = null.

if (***CurrentAccuracyStatus*** is null)

Locate any record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the TestResult is equal to "PASSED", "EXC168H", "INPROG", or "FAILED" and the quarter is prior to the quarter of the ***CurrentOperatingDate*** and the quarter is subsequent to the quarter of the later of the ***PriorAccuracyRecord***.EndDate and the ***PriorAccuracyRecord***.ReinstallationDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

Set *ValidFuelFlowTestExistsForEachComponent* = true.
Set *CertificationCheckDate* = the later of the ***PriorAccuracyRecord***.EndDate and the ***PriorAccuracyRecord***.ReinstallationDate.

for each *ComponentRecord* in ***FuelFlowComponentRecords***

Locate the latest *AccuracyTestRecord* for the location in ***AccuracyTestRecordsByLocationForQAStatus*** where:

1) ComponentID is equal to the *ComponentRecord*.ComponentID.
2) The quarter of the later of the EndDate and the ReinstallationDate is before the quarter of ***CurrentOperatingDate***.

if (*AccuracyTestRecord* was NOT found)

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else if (*AccuracyTestRecord*.TestResultCode does NOT equal "PASSED")

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else if (the later of *AccuracyTestRecord*.EndDate and the *AccuracyTestRecord*.ReinstallationDate is on or after ***CurrentFuelFlowRecord***.SystemBeginDate)

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else

Set *CertificationCheckDate* = the later of *CertificationCheckDate* AND *AccuracyTestRecord*.EndDate AND *AccuracyTestRecord*.ReinstallationDate.

if (*AccuracyTestRecord*.TestTypeCode == "FFACCTT")

Locate the latest *PeiTestRecord* for the location in ***PEITestRecordsByLocationForQAStatus*** where:

1) ComponentID is equal to the *ComponentRecord*.ComponentID.
2) The quarter of the EndDate is before the quarter of ***CurrentOperatingDate***.

if (*PeiTestRecord* was NOT found)

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else if (*PeiTestRecord*.TestResultCode does NOT equal "PASSED")

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else if (*PeiTestRecord*.EndDate is on or after ***CurrentFuelFlowRecord***.SystemBeginDate)

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else

Set *CertificationCheckDate* = the later of *CertificationCheckDate* AND *PeiTestRecord*.EndDate.

if (*ValidFuelFlowTestExistsForEachComponent* is equal to true) AND (*CertificationCheckDate* is NOT null)

Set ***FF2LAccuracyEligible*** = true.
Set ***FF2LAccuracyCheckDate*** = *CertificationCheckDate.*

else

Set ***FF2LAccuracyEligible*** = true.
Set ***FF2LAccuracyCheckDate*** = the later of the ***PriorAccuracyRecord***.EndDate and the ***PriorAccuracyRecord***.ReinstallationDate.

for each record in ***FuelFlowComponentRecords***

if (***FuelFlowComponentRecords***.ComponentID is not equal to ***FuelFlowComponentRecordToCheck***.ComponentID)

Locate the latest record in ***AccuracyTestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecords***.ComponentID and the TestResult is equal to "PASSED" and the quarter of the later of the EndDate and the ReinstallationDate is in the same or adjacent quarter of the later of the ***PriorAccuracyRecord***.EndDate and the ***PriorAccuracyRecord***.ReinstallationDate.

if (***AccuracyTestRecordsByLocationForQAStatus*** is not found)

Set ***FF2LAccuracyEligible*** = false, and exit this check.

else

Set ***FF2LAccuracyCheckDate*** = the later of ***FF2LAccuracyCheckDate*** and the ***AccuracyTestRecordsByLocationForQAStatus***.EndDate and the ***AccuracyTestRecordsByLocationForQAStatus***.ReinstallationDate.

if (***AccuracyTestRecordsByLocationForQAStatus***.TestTypeCode is equal to "FFACCTT")

Locate the latest record in ***PEITestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecords***.ComponentID and the TestResult is equal to "PASSED" and the quarter is in the same or adjacent quarter of the later of the ***PriorAccuracyRecord***.EndDate and the

PriorAccuracyRecord.ReinstallationDate.

if (***PEITestRecordsByLocationForQAStatus*** is not found)

Set ***FF2LAccuracyEligible*** = false, and exit this check.

else

Set ***FF2LAccuracyCheckDate*** = the later of ***FF2LAccuracyCheckDate*** and the ***PEITestRecordsByLocationForQAStatus***.EndDate.

else if (***PriorAccuracyTestRecord***.TestTypeCode is equal to "FFACCTT")

Locate the latest record in ***PEITestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecords***.ComponentID and the TestResult is equal to "PASSED" and the quarter is in the same or adjacent quarter of the ***PriorAccuracyRecord***.EndDate.

if (***PEITestRecordsByLocationForQAStatus*** is not found)

Set ***FF2LAccuracyEligible*** = false, and exit this check.

else

Set ***FF2LAccuracyCheckDate*** = the later of ***FF2LAccuracyCheckDate*** and the ***PEITestRecordsByLocationForQAStatus***.EndDate.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Code: ADESTAT-5

Check Name: Evaluate Fuel Flow to Load Tests (Accuracy)

Related Former Checks:

Applicability:

Description: Evaluates the Fuel flow to load tests for a flow meter that is eligible to use fuel flow to load tests.

Specifications:

Set ***FF2LAccuracyBeginYearQuarter*** = null.
Set ***FF2LAccuracyEndYearQuarter*** = null.
Set ***InvalidFF2LTestNumber*** = null.
Set ***MissingFF2LYearQuarter*** = null.

if (***FF2LAccuracyEligible*** == true)

Locate any record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the TestResult is equal to "FAILED" and the quarter is prior to the quarter of the ***CurrentOperatingDate*** and the quarter is subsequent to the quarter of the later of the ***PriorAccuracyRecord***.EndDate and the ***PriorAccuracyRecord***.ReinstallationDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber
Set ***CurrentAccuracyStatus*** = "OOC-Fuel Flow to Load Test Failed".
exit check.

Locate any record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the TestResult is NULL and the quarter is prior to the quarter of the ***CurrentOperatingDate*** and the quarter is subsequent to the quarter of the later of the ***PriorAccuracyRecord***.EndDate and the ***PriorAccuracyRecord***.ReinstallationDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber
Set ***CurrentAccuracyStatus*** = "OOC-Fuel Flow to Load Test Has Critical Errors".
exit check.

Locate any record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the QANeedsEvaluation flag is equal to "Y" and the quarter is prior to the quarter of the ***CurrentOperatingDate*** and the quarter is subsequent to the quarter of the later of the ***PriorAccuracyRecord***.EndDate and the ***PriorAccuracyRecord***.ReinstallationDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber
Set ***CurrentAccuracyStatus*** = "Fuel Flow to Load Test Has Not Yet Been Evaluated".
exit check.

Locate the earliest record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the TestResult is equal to "PASSED", "FEW168H", or "EXC168H" and the quarter is prior to the quarter of the ***CurrentOperatingDate*** and the quarter is subsequent to the quarter of the later of the ***PriorAccuracyRecord***.EndDate and the ***PriorAccuracyRecord***.ReinstallationDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

Set ***FF2LAccuracyBeginYearQuarter*** = ***FF2LTestRecordsByLocationForQAStatus***.Year &
FF2LTestRecordsByLocationForQAStatus.Quarter.

else

Set ***FF2LAccuracyBeginYearQuarter*** = null

Locate the latest record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the TestResult is equal to "INPROG" and the quarter is prior to the quarter of the ***CurrentOperatingDate*** and the quarter is subsequent to the quarter of the later of the ***PriorAccuracyRecord***.EndDate and the ***PriorAccuracyRecord***.ReinstallationDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

if (***FF2LAccuracyBeginYearQuarter*** is not null AND ***FF2LTestRecordsByLocationForQAStatus***.Year/Quarter > ***FF2LAccuracyBeginYearQuarter***)

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber
Set ***CurrentAccuracyStatus*** = "OOC-Invalid Fuel Flow to Load Test".
exit check.

else if (***FF2LTestRecordsByLocationForQAStatus***.Year/Quarter is more than 4 quarters after the quarter of the ***FF2LAccuracyCheckDate***)

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber
Set ***CurrentAccuracyStatus*** = "Undetermined-Baseline Period Expired".
exit check.

Set ***FF2LAccuracyBeginYearQuarter*** = the quarter after the quarter of the ***FF2LAccuracyCheckDate.***

Locate the latest record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the TestResult is equal to "PASSED", "EXC168H", "INPROG", or "FEW168H", and the quarter is prior to the quarter of the ***CurrentOperatingDate*** and the quarter is subsequent to the quarter of the later of the ***PriorAccuracyRecord***.EndDate and the ***PriorAccuracyRecord***.ReinstallationDate.

Set ***FF2LAccuracyEndYearQuarter*** = ***FF2LTestRecordsByLocationForQAStatus***.Year & ***FF2LTestRecordsByLocationForQAStatus***.Quarter.

for each quarter between the ***FF2LAccuracyBeginYearQuarter*** and the ***FF2LAccuracyEndYearQuarter*** (inclusive)

Locate any record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the quarter is equal to the quarter to check.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

if (***FF2LTestRecordsByLocationForQAStatus***.TestResult = "FEW168H")

Locate a record in ***OperatingSuppDataRecordsbyLocation*** where the FuelCode is equal to ***CurrentFuelFlowRecord.***FuelCode and the OpTypeCode is equal to "OPHOURS" or "OSHOURS" and the quarter is equal to the quarter to check and the OpValue >= 168.

if (***OperatingSuppDataRecordsbyLocation*** is found)

Locate any record in ***FF2LBaselineRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID, and the EndDate is within the quarter being checked.

If not found,

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber

Set ***CurrentAccuracyStatus*** = "OOC-Invalid Fuel Flow to Load Test".
Exit check.

else

Locate a record in ***OperatingSuppDataRecordsbyLocation*** where the FuelCode is equal to ***CurrentFuelFlowRecord.***FuelCode and the OpTypeCode is equal to "OPHOURS" or "OSHOURS" and the quarter is equal to the quarter to check and the OpValue >= 168.

if (***OperatingSuppDataRecordsbyLocation*** is found)

Set ***MissingFF2LYearQuarter*** equal to the year/quarter to check.
Set ***CurrentAccuracyStatus*** = "Undetermined-Missing Fuel Flow to Load Test".
Exit check.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Code: ADESTAT-6

Check Name: Determine Accuracy Test Expiration Date

Related Former Checks:

Applicability:

Description: Determines the expiration date for the prior applicable Accuracy Test.

Specifications:

Set ***AccuracyMissingOpDataInfo*** = null.

if (***CurrentAccuracyStatus*** is null)

Set *PriorTestExpirationDate* = ***PriorAccuracyRecord***.TestExpirationDate.

for each quarter subsequent to the quarter of the later of the ***PriorAccuracyRecord***.EndDate/Hour and the ***PriorAccuracyRecord***.ReinstallationDate/Hour and prior to the quarter of the ***CurrentOperatingDate/Hour***

Set *OSO Reporter* to false.

Locate a ***LocationReportingFrequency*** record for the test location where ReportingFrequencyCode = "OS", the BeginQuarter is on or before the quarter being checked, and the EndQuarter is null or is on or after the quarter being checked.

If found,

Set *OSO Reporter* to true.

if (*OSO Reporter* == false or the quarter to check is the 3rd quarter)

if (***FF2LAccuracyEligible*** == true and the quarter to check is between the ***FF2LAccuracyBeginYearQuarter*** and the ***FF2LAccuracyEndYearQuarter*** (inclusive))

Add 1 quarter to the *PriorTestExpirationDate.*

else

if (***EarliestLocationReportDate*** > the last day of the quarter being checked)

Add 1 quarter to the *PriorTestExpirationDate.*

else

Locate the record in ***OperatingSuppDataRecordsbyLocation*** where the OpTypeCode is equal to "OPHOURS" and the reporting period is equal to the quarter to check and the FuelCode is equal to ***CurrentFuelFlowRecord.***FuelCode.

if (***OperatingSuppDataRecordsbyLocation*** is not found)

Locate the record in ***OperatingSuppDataRecordsbyLocation*** where the OpTypeCode is equal to "OPHOURS" and the reporting period is equal to the quarter to check and the FuelCode is null.

if (***OperatingSuppDataRecordsbyLocation*** is not found)

Set ***AccuracyMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked).

else

Add 1 quarter to the *PriorTestExpirationDate*.

else if (***OperatingSuppDataRecordsbyLocation***.OpValue <= 168)

Add 1 quarter to the *PriorTestExpirationDate*.

else

Locate a record in ***TestExtensionExemptionRecords*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecordToCheck***.ComponentID, the reporting period is equal to the quarter to check , AND the ExtensionExemptionCode is equal to "NONQADB".

if (***TestExtensionExemptionRecords*** is found)

Add 1 quarter to the *PriorTestExpirationDate*.

else if (*OSO Reporter* == true and the quarter to check is the 2nd quarter)

if (***FF2LAccuracyEligible*** == true and the quarter to check is between the ***FF2LAccuracyBeginYearQuarter*** and the ***FF2LAccuracyEndYearQuarter*** (inclusive))

Add 1 quarter to the *PriorTestExpirationDate*.

else

Locate a record in ***TestExtensionExemptionRecords*** for the location where the reporting period is equal to the quarter to check , AND the ExtensionExemptionCode is equal to "NONQAOS" and the FuelCode is equal to the ***CurrentFuelFlowRecord.***FuelCode.

if (***TestExtensionExemptionRecords*** is found)

Add 1 quarter to the *PriorTestExpirationDate*.

else

Locate a record in ***TestExtensionExemptionRecords*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecordToCheck***.ComponentID, the reporting period is equal to the quarter to check , AND the ExtensionExemptionCode is equal to "NONQADB".

if (***TestExtensionExemptionRecords*** is found)

Add 1 quarter to the *PriorTestExpirationDate*.

else if (*OSO Reporter* == true and the quarter to check is the 1st or 4th quarter)

if (***FF2LAccuracyEligible*** == true and the 2nd quarter following the quarter being checked is between the ***FF2LAccuracyBeginYearQuarter*** and the ***FF2LAccuracyEndYearQuarter*** (inclusive))

Add 1 quarter to the *PriorTestExpirationDate*.

else

Locate a record in ***TestExtensionExemptionRecords*** for the location where the reporting period is equal to the quarter to check , AND the ExtensionExemptionCode is equal to "NONQAOS" and the FuelCode

is equal to the ***CurrentFuelFlowRecord.***FuelCode.

if (***TestExtensionExemptionRecords*** is found)

Add 1 quarter to the *PriorTestExpirationDate.*

else

Locate a record in ***TestExtensionExemptionRecords*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecordToCheck***.ComponentID, the reporting period is equal to the quarter to check , AND the ExtensionExemptionCode is equal to "NONQADB".

if (***TestExtensionExemptionRecords*** is found)

Add 1 quarter to the *PriorTestExpirationDate.*

if (***CurrentOperatingDate/Hour*** > *PriorTestExpirationDate*)

if (***AccuracyMissingOpDataInfo*** is not null)

Set ***CurrentAccuracyStatus*** = "Missing Op Data"
Return result ***CurrentAccuracyStatus*** .

else if (***FF2LAccuracyEligible*** == false)

Set ***CurrentAccuracyStatus*** = "OOC-Accuracy Test Expired-Fuel Flow To Load Test Ignored".
Return result ***CurrentAccuracyStatus*** .

else

Set ***CurrentAccuracyStatus*** = "OOC-Accuracy Test Expired"

else

Set ***CurrentAccuracyStatus*** = "IC-Extension"

If (***CurrentAccuracyStatus*** does not begin with "IC" and is not null)

if (***CurrentAccuracyStatus*** starts with "OOC" or "Undetermined" AND ***InvalidAccuracyRecord*** is not null)

Set ***CurrentAccuracyStatus*** = ***CurrentAccuracyStatus*** & "*"

Return result ***CurrentAccuracyStatus***.

else if (***InapprorpriateTransmitterTransducerTest*** == true)

Return result "Inappropriate Transmitter Transducer Test" // do NOT set current accuracy status

Results:

Result	Response	Severity
Accuracy Test Not Yet Evaluated	The [testtype] status for [key] could not be determined, because the applicable prior [testtype] with TestNumber [testnum] has not yet been evaluated.	Critical Error Level 1
Fuel Flow to Load Test Has Not Yet Been Evaluated	The [testtype] status for [key] could not be determined, because a prior fuel-flow-to-load test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum] has not yet been evaluated.	Critical Error Level 1
Inappropriate Transmitter Transducer Test	The prior [testtype] for [key] with TestNumber [testnum] is a transmitter/transducer test], but this type of test is inappropriate for the SampleAcquisitionMethodCode for the fuel flowmeter. A transmitter/transducer test can only be performed on a NOZ, VEN, and ORF fuel flowmeter.	Critical Error Level 2
Missing Op Data	The [testtype] status for [key] could not be determined, because the Op Supp Data record for OPHOURS, OSHOURS, or OPDAYS is missing for [MISSINGOPDATAINFO] (and possibly other previous reporting periods). If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
OOC-Accuracy Test Aborted	The applicable prior [testtype] for [key] with TestNumber [testnum] was aborted.	Critical Error Level 1
OOC-Accuracy Test Aborted*	The prior [testtype] for [key] with TestNumber [testnum] was aborted. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Accuracy Test Expired	The prior [testtype] for [key] with TestNumber [testnum] has expired.	Critical Error Level 1
OOC-Accuracy Test Expired*	The prior [testtype] for [key] with TestNumber [testnum] has expired. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Accuracy Test Expired-Fuel Flow To Load Test Ignored	The prior [testtype] for [key] with TestNumber [testnum] has expired. A prior fuel-flow-to-load test for MonitoringSystemID [ID] was ignored.	Critical Error Level 1
OOC-Accuracy Test Expired-Fuel Flow To Load Test Ignored*	The prior [testtype] for [key] with TestNumber [testnum] has expired. A prior fuel-flow-to-load test for MonitoringSystemID [ID] was ignored. An invalid prior [testtype] with TestNumber [invtestnum] was also ignored.	Critical Error Level 1
OOC-Accuracy Test Failed	The applicable prior [testtype] for [key] with TestNumber [testnum] failed.	Critical Error Level 1
OOC-Accuracy Test Failed*	The prior [testtype] for [key] with TestNumber [testnum] failed. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Accuracy Test Has Critical Errors	The applicable prior [testtype] for [key] with TestNumber [testnum] has critical errors.	Critical Error Level 1
OOC-Accuracy Test Has Critical Errors*	The prior [testtype] for [key] with TestNumber [testnum] has critical errors. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Event	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for [key], but you did not perform a subsequent [testtype].	Critical Error Level 1
OOC-Event*	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for [key], but you did not perform a subsequent [testtype]. An invalid [testtype] was ignored.	Critical Error Level 1
OOC-Fuel Flow to Load Test Failed	The [testtype] status for [key] could not be determined, because a prior fuel-flow-to-load test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum] has failed.	Critical Error Level 1
OOC-Fuel Flow to Load Test Failed*	The [testtype] status for [key] could not be determined, because a prior fuel-flow-to-load test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum] has failed. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1

OOC-Fuel Flow to Load Test Has Critical Errors	The [testtype] status for [key] could not be determined, because a prior fuel-flow-to-load test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum] has critical errors.	Critical Error Level 1
OOC-Fuel Flow to Load Test Has Critical Errors*	The [testtype] status for [key] could not be determined, because a prior fuel-flow-to-load test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum] has critical errors. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Invalid Fuel Flow to Load Test	The [testtype] status for [key] could not be determined, because one or more prior fuel-flow-to-load tests, including the test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum], are invalid. These tests may be invaild because (1) the TestResultCode indicates that baseline data collection is ongoing, yet you reported a prior test indicating that baseline data collection was completed; or (2) the TestResultCode indicates that there were fewer than 168 fuel QA operating hours in the quarter, yet your emissions data for that quarter indicates otherwise.	Critical Error Level 1
OOC-Invalid Fuel Flow to Load Test*	The [testtype] status for [key] could not be determined, because one or more prior fuel-flow-to-load tests, including the test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum], are invalid. These tests may be invaild because (1) the TestResultCode indicates that baseline data collection is ongoing, yet you reported a prior test indicating that baseline data collection was completed; or (2) the TestResultCode indicates that there were fewer than 168 fuel QA operating hours in the quarter, yet your emissions data for that quarter indicates otherwise. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-No Prior Accuracy Test	You did not report a prior [testtype] for [key].	Critical Error Level 1
OOC-No Prior Accuracy Test*	You did not report a valid prior [testtype] for [key]. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
Undetermined-Baseline Period Expired	The [testtype] status for [key] could not be determined, because, according to the fuel-flow-to-load test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum], baseline data was still being collected after the deadline.	Critical Error Level 2
Undetermined-Missing Fuel Flow to Load Test	The [testtype] status for [key] could not be determined, because a prior fuel-flow-to-load test for MonitoringSystemID [ID] was missing for [missingff2l].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Code: ADESTAT-7

Check Name: Determine if Component Requires a PEI Test

Related Former Checks:

Applicability:

Description: Determines a if an Appendix D fuel flow meter requires a PEI test.

Specifications:

Set ***PEIRequired*** = false.

if (***PriorAccuracyRecord*** is not null and ***PriorAccuracyRecord***.TestTypeCode is equal to "FFACCTT")
Set ***PEIRequired*** = true.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Code: ADESTAT-8

Check Name: Locate Most Recent Prior PEI Test

Related Former Checks:

Applicability:

Description: Determines if there is an applicable prior PEI test.

Specifications:

Set ***CurrentPEIStatus*** = null.
Set ***PriorPEIRecord*** = null.

if (***PEIRequired*** == true)

Locate the most recent record in ***PEITestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecordToCheck***.ComponentID and the EndDate/Hour is prior to the ***CurrentOperatingDate/Hour***

if (***PEITestRecordsByLocationForQAStatus*** is found)

Set ***PriorPEIRecord*** = the found record in ***PEITestRecordsByLocationForQAStatus*** .

if (***PriorPEIRecord***.QANeedsEvaluationFlag = "Y")

Set ***CurrentPEIStatus*** = "PEI Test Not Yet Evaluated".

else if (***PriorPEIRecord***.TestResultCode is null)

Set ***CurrentPEIStatus*** = "OOC-PEI Test Has Critical Errors".

else if (***PriorPEIRecord***.TestResultCode = "FAILED")

Set ***CurrentPEIStatus*** = "OOC-PEI Test Failed".

else if (***PriorPEIRecord***.TestResultCode = "ABORTED")

Set ***CurrentPEIStatus*** = "OOC-PEI Test Aborted".

else

Set ***CurrentPEIStatus*** = "OOC-No Prior PEI Test".

if (***CurrentPEIStatus*** is not null)

Return result ***CurrentPEIStatus***.

Results:

Result	Response	Severity
OOC-No Prior PEI Test	You did not report a prior [testtype] for [key].	Critical Error Level 1
OOC-PEI Test Aborted	The applicable prior [testtype] for [key] with TestNumber [testnum] was aborted.	Critical Error Level 1
OOC-PEI Test Failed	The applicable prior [testtype] for [key] with TestNumber [testnum] failed.	Critical Error Level 1
OOC-PEI Test Has Critical Errors	The applicable prior [testtype] for [key] with TestNumber [testnum] has critical errors.	Critical Error Level 1
PEI Test Not Yet Evaluated	The [testtype] status for [key] could not be determined, because the applicable prior [testtype] with TestNumber [testnum] has not yet been evaluated.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Code: ADESTAT-9

Check Name: Locate Most Recent Prior PEI Event

Related Former Checks:

Applicability:

Description: Determines if there is a applicable prior event requiring an PEI test.

Specifications:

Set ***PriorPEIEventRecord*** = null.

If (***PEIRequired*** == true AND ***CurrentPEIStatus*** is null)

Locate the most recent record in ***QACertificationEventRecords*** WHERE

a) the ComponentID is equal to the ***FuelFlowComponentRecordToCheck***.ComponentID AND
b) PEIRequired is equal to "Y" AND
c) the QACertEventDate/Hour is prior to the ***CurrentOperatingDate/Hour*** AND
d) the QACertEventDate/Hour is after the ***PriorPEIRecord***.EndDate/Hour.

if (***QACertificationEventRecords*** is found)

Set ***PriorPEIEventRecord*** = the found record in ***QACertificationEventRecords***.
Set ***CurrentPEIStatus*** = "OOC-Event".

else

Set *PriorTestExpirationDate* = 5 years (20 calendar quarters) after the end of the quarter of the ***PriorPEIRecord***.EndDate.

if (***CurrentOperatingDate*** is AFTER the *PriorTestExpirationDate*)

Set ***CurrentPEIStatus*** = "OOC-PEI Test Expired".

else

Set *PriorTestExpirationDate* = 12 quarters after the end of the quarter of the ***PriorPEIRecord***.EndDate.
Set ***PriorPEIRecord***.TestExpirationDate = *PriorTestExpirationDate.*

If (***CurrentOperatingDate*** is ON OR BEFORE the *PriorTestExpirationDate*)

Set ***CurrentPEIStatus*** = "IC".

If (***CurrentPEIStatus*** starts with "OOC")

Return result ***CurrentPEIStatus***.

Results:

Result	Response	Severity
OOC-Event	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for [key], but you did not perform a subsequent [testtype].	Critical Error Level 1
OOC-PEI Test Expired	The prior [testtype] for [key] with TestNumber [testnum] has expired.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Code: ADESTAT-10

Check Name: Determine Eligibility for Fuel Flow to Load Testing (PEI)

Related Former Checks:

Applicability:

Description: Determines if this component is eligible to extend their PEI text expiration date using fuel flow to load testing.

Specifications:

Set ***FF2LPEIEligible*** = null
Set ***FF2LPEICheckDate*** = null.

if (***PEIRequired*** == true AND ***CurrentPEIStatus*** is null)

Locate any record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the CalculatedTestResult is equal to "PASSED", "FEW168H", "EXC168H", "INPROG", or "FAILED" and the Year/Quarter is prior to the quarter of the ***CurrentOperatingDate*** and the Year/Quarter is subsequent to the quarter of the ***PriorPEIRecord***.EndDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

Set *ValidFuelFlowTestExistsForEachComponent* = true.
Set *CertificationCheckDate* = the later of the ***PriorAccuracyRecord***.EndDate and the ***PriorAccuracyRecord***.ReinstallationDate.

for each *ComponentRecord* in ***FuelFlowComponentRecords***

Locate the latest *AccuracyTestRecord* for the location in ***AccuracyTestRecordsByLocationForQAStatus*** where:

1) ComponentID is equal to the *ComponentRecord*.ComponentID.
2) The quarter of the later of the EndDate and the ReinstallationDate is before the quarter of ***CurrentOperatingDate***.

if (*AccuracyTestRecord* was NOT found)

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else if (*AccuracyTestRecord*.TestResultCode does NOT equal "PASSED")

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else if (the later of *AccuracyTestRecord*.EndDate and the *AccuracyTestRecord*.ReinstallationDate is on or after ***CurrentFuelFlowRecord***.SystemBeginDate)

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else

Set *CertificationCheckDate* = the later of *CertificationCheckDate* AND *AccuracyTestRecord*.EndDate AND *AccuracyTestRecord*.ReinstallationDate.

if (*AccuracyTestRecord*.TestTypeCode == "FFACCTT")

Locate the latest *PeiTestRecord* for the location in ***PEITestRecordsByLocationForQAStatus*** where:

1) ComponentID is equal to the *ComponentRecord*.ComponentID.
2) The quarter of the EndDate is before the quarter of ***CurrentOperatingDate***.

if (*PeiTestRecord* was NOT found)

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else if (*PeiTestRecord*.TestResultCode does NOT equal "PASSED")

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else if (*PeiTestRecord*.EndDate is on or after ***CurrentFuelFlowRecord***.SystemBeginDate)

Set *ValidFuelFlowTestExistsForEachComponent* = false.

else

Set *CertificationCheckDate* = the later of *CertificationCheckDate* AND *PeiTestRecord*.EndDate.

if (*ValidFuelFlowTestExistsForEachComponent* is equal to true) AND (*CertificationCheckDate* is NOT null)

Set ***FF2LPEIEligible*** = true.
Set ***FF2LPEICheckDate*** = *CertificationCheckDate*.

else

Set ***FF2LPEIEligible*** = true.
Set ***FF2LPEICheckDate*** = ***PriorPEIRecord***.EndDate.

for each record in ***FuelFlowComponentRecords***

Locate the latest record in ***AccuracyTestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecords***.ComponentID and the TestResult is equal to "PASSED" and the Year/Quarter of the later of the EndDate and Reinstallation Date is in the same or adjacent quarter of the ***PriorPEIRecord***.EndDate.

if (***AccuracyTestRecordsByLocationForQAStatus*** is not found)
Set ***FF2LPEIEligible*** = false, and exit check.

else

Set ***FF2LPEICheckDate*** = the later of ***FF2LPEICheckDate*** and the ***AccuracyTestRecordsByLocationForQAStatus***.EndDate and the ***AccuracyTestRecordsByLocationForQAStatus***.ReinstallationDate.

if (***FuelFlowComponentRecords***.ComponentID is not equal to ***FuelFlowComponentRecordToCheck***.ComponentID AND ***AccuracyTestRecordsByLocationForQAStatus***.TestTypeCode is equal to "FFACCTT")

Locate the latest record in ***PEITestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***FuelFlowComponentRecords***.ComponentID and the TestResult is equal to "PASSED" and the Year/Quarter is in the same or adjacent quarter of the ***PriorPEIRecord***.EndDate.

if (***PEITestRecordsByLocationForQAStatus*** is not found)
Set ***FF2LPEIEligible*** = false, and exit check.

else
Set ***FF2LPEICheckDate*** = the later of ***FF2LPEICheckDate*** and the ***PEITestRecordsByLocationForQAStatus***.EndDate.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Code: ADESTAT-11

Check Name: Evaluate Fuel Flow to Load Tests (PEI)

Related Former Checks:

Applicability:

Description: Evaluates the Fuel flow to load tests for a flow meter that is eligible to use fuel flow to load tests.

Specifications:

Set ***FF2LPEIBeginYearQuarter*** = null.
Set ***FF2LPEIEndYearQuarter*** = null.

if (***FF2LPEIEligible*** == true)

Locate any record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the TestResultCode is equal to "FAILED" and the Year/Quarter is prior to the quarter of the ***CurrentOperatingDate*** and the Year/Quarter is subsequent to the quarter of the ***PriorPEIRecord***.EndDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

Locate a record in ***QACertificationEventRecords*** WHERE

a) the ComponentID is equal to the ***FuelFlowComponentRecordToCheck***.ComponentID AND
b) the QACertEventCode is equal to "410"
c) the RequiredTestCode is equal to "53"AND
d) the QACertEventDate/Hour is after the EndDate/Hour and of the located failed ***FF2LTestRecordsByLocationForQAStatus*** record.
e) the QACertEventDate/Hour is prior to the ***CurrentOperatingDate/Hour***

if (***QACertificationEventRecords*** is not found)

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber
Set ***CurrentPEIStatus*** = "OOC-Fuel Flow to Load Test Failed".
Return result ***CurrentPEIStatus***.

Locate any record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the TestResultCode is NULL and the Year/Quarter is prior to the quarter of the ***CurrentOperatingDate*** and the Year/Quarter is subsequent to the quarter of the ***PriorPEIRecord***.EndDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber
Set ***CurrentPEIStatus*** = "OOC-Fuel Flow to Load Test Has Critical Errors".
Return result ***CurrentPEIStatus***.

Locate any record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the QANeedsEvaluation flag is equal to "Y" and the Year/Quarter is prior to the quarter of the ***CurrentOperatingDate*** and the Year/Quarter is subsequent to the quarter of the ***PriorPEIRecord***.EndDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber
Set ***CurrentPEIStatus*** = "Fuel Flow to Load Test Has Not Yet Been Evaluated".
Return result ***CurrentPEIStatus***.

Locate the earliest record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the TestResultCode is equal to "PASSED", "FEW168H", or "EXC168H" and the Year/Quarter is prior to the quarter of the ***CurrentOperatingDate*** and the Year/Quarter is subsequent to the quarter of the

PriorPEIRecord.EndDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

Set ***FF2LPEIBeginYearQuarter*** = ***FF2LTestRecordsByLocationForQAStatus***.Year & ***FF2LTestRecordsByLocationForQAStatus***.Quarter.

else

Set ***FF2LPEIBeginYearQuarter*** = null

Locate the latest record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the TestResultCode is equal to "INPROG" and the Year/Quarter is prior to the quarter of the ***CurrentOperatingDate*** and the Year/Quarter is subsequent to the quarter of the ***PriorPEIRecord***.EndDate.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

if (Set ***FF2LPEIBeginYearQuarter*** is not null AND ***FF2LTestRecordsByLocationForQAStatus***.Year/Quarter > ***FF2LPEIBeginYearQuarter***)

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber
Set ***CurrentPEIStatus*** = "OOC-Invalid Fuel Flow to Load Test".
Return result ***CurrentPEIStatus***.

else if (***FF2LTestRecordsByLocationForQAStatus***.Year/Quarter is more than 4 quarters after the quarter of the ***FF2LPEICheckDate***)

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber
Set ***CurrentPEIStatus*** = "Undetermined-Baseline Period Expired".
Return result ***CurrentPEIStatus***.

Set ***FF2LPEIBeginYearQuarter*** to the quarter after the quarter of the ***FF2LPEICheckDate***.

Locate the latest record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the TestResultCode is equal to "PASSED", "FEW168H", or "EXC168H" and the Year/Quarter is prior to the quarter of the ***CurrentOperatingDate*** and the Year/Quarter is subsequent to the quarter of the ***PriorPEIRecord***.EndDate.

Set ***FF2LPEIEndYearQuarter*** = ***FF2LTestRecordsByLocationForQAStatus***.Year & ***FF2LTestRecordsByLocationForQAStatus***.Quarter.

for each quarter between the ***FF2LPEIBeginYearQuarter*** and the ***FF2LPEIEndYearQuarter*** (inclusive)

Locate any record in ***FF2LTestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID and the Year/Quarter is equal to the year/quarter to check.

if (***FF2LTestRecordsByLocationForQAStatus*** is found)

if (***FF2LTestRecordsByLocationForQAStatus***.CalculatedTestResult = "FEW168H")

Locate a record in ***OperatingSuppDataRecordsbyLocation*** where the FuelCode is equal to ***CurrentFuelFlowRecord***.FuelCode and the OpTypeCode is equal to "OPHOURS" or "OSHOURS" and the Year/Quarter is equal to the year/quarter to check and the OpValue >= 168.

if (***OperatingSuppDataRecordsbyLocation*** is found)

Locate any record in ***FF2LBaselineRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***FuelFlowComponentRecordToCheck***.SystemID, and the EndDate is

within the quarter being checked.

If not found,

Set ***InvalidFF2LTestNumber*** = ***FF2LTestRecordsByLocationForQAStatus***.TestNumber
Set ***CurrentPEIStatus*** = "OOC-Invalid Fuel Flow to Load Test".
Return result ***CurrentPEIStatus***.

else

Locate a record in ***OperatingSuppDataRecordsbyLocation*** where the FuelCode is equal to ***CurrentFuelFlowRecord.***FuelCode and the OpTypeCode is equal to "OPHOURS" or "OSHOURS" and the Year/Quarter is equal to the year/quarter to check and the OpValue >= 168.

if (***OperatingSuppDataRecordsbyLocation*** is found)

Set ***MissingFF2LYearQuarter*** equal to the year/quarter to check.
Set ***CurrentPEIStatus*** = "Undetermined-Missing Fuel Flow to Load Test".
Return result ***CurrentPEIStatus***.

Results:

Result	Response	Severity
Fuel Flow to Load Test Has Not Yet Been Evaluated	The [testtype] status for [key] could not be determined, because a prior fuel-flow-to-load test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum] has not yet been evaluated.	Critical Error Level 1
OOC-Fuel Flow to Load Test Failed	The [testtype] status for [key] could not be determined, because a prior fuel-flow-to-load test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum] has failed.	Critical Error Level 1
OOC-Fuel Flow to Load Test Has Critical Errors	The [testtype] status for [key] could not be determined, because a prior fuel-flow-to-load test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum] has critical errors.	Critical Error Level 1
OOC-Invalid Fuel Flow to Load Test	The [testtype] status for [key] could not be determined, because one or more prior fuel-flow-to-load tests, including the test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum], are invalid. These tests may be invaild because (1) the TestResultCode indicates that baseline data collection is ongoing, yet you reported a prior test indicating that baseline data collection was completed; or (2) the TestResultCode indicates that there were fewer than 168 fuel QA operating hours in the quarter, yet your emissions data for that quarter indicates otherwise.	Critical Error Level 1
Undetermined-Baseline Period Expired	The [testtype] status for [key] could not be determined, because, according to the fuel-flow-to-load test for MonitoringSystemID [ID] with TestNumber [ff2ltestnum], baseline data was still being collected after the deadline.	Critical Error Level 2
Undetermined-Missing Fuel Flow to Load Test	The [testtype] status for [key] could not be determined, because a prior fuel-flow-to-load test for MonitoringSystemID [ID] was missing for [missingff2l].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Code: ADESTAT-12

Check Name: Determine PEI Test Expiration Date

Related Former Checks:

Applicability:

Description: Determines the expiration date for the prior applicable PEITest.

Specifications:

Set ***PEIMissingOpDataInfo*** = null.

if (***PEIRequired*** == true AND ***CurrentPEIStatus*** is null)

Set *PriorTestExpirationDate* = ***PriorPEIRecord***.TestExpirationDate.

for each quarter subsequent to the quarter of the ***PriorPEIRecord***.EndDate and prior to the quarter of the ***CurrentOperatingDate***

Set *OSO Reporter* to false.

Locate a ***LocationReportingFrequency*** record for the test location where ReportingFrequencyCode = "OS", the BeginQuarter is on or before the quarter being checked, and the EndQuarter is null or is on or after the quarter being checked.

If found,

Set *OSO Reporter* to true.

if (*OSO Reporter* == false or the quarter to check is the 3rd quarter)

if (***FF2LPEIEligible*** == true and the quarter to check is between the ***FF2LPEIBeginYearQuarter*** and the ***FF2LPEIEndYearQuarter*** (inclusive))

Add 1 quarter to the *PriorTestExpirationDate.*

else if (*OSO Reporter* == true and the quarter to check is the 2nd quarter)

if (***FF2LPEIEligible*** == true and the quarter to check is between the ***FF2LPEIBeginYearQuarter*** and the ***FF2LPEIEndYearQuarter*** (inclusive))

Add 3 quarters to the *PriorTestExpirationDate.*

if (***CurrentOperatingDate*** > *PriorTestExpirationDate*)

if (***PEIMissingOpDataInfo*** is not null)

Set ***CurrentPEIStatus*** = "Missing Op Data"

else if (***FF2LPEIEligible*** == false)

Set ***CurrentPEIStatus*** = "OOC-PEI Test Expired-Fuel Flow To Load Test Ignored"

else

Set ***CurrentPEIStatus*** = "OOC-PEI Test Expired"

Return result ***CurrentPEIStatus***.

else

Set ***CurrentPEIStatus*** = "IC-Extension"

Results:

Result	Response	Severity
Missing Op Data	The [testtype] status for [key] could not be determined, because the Op Supp Data record for OPHOURS, OSHOURS, or OPDAYS is missing for [MISSINGOPDATAINFO] (and possibly other previous reporting periods). If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
OOC-PEI Test Expired	The prior [testtype] for [key] with TestNumber [testnum] has expired.	Critical Error Level 1
OOC-PEI Test Expired-Fuel Flow To Load Test Ignored	The prior [testtype] for [key] with TestNumber [testnum] has expired. A prior fuel-flow-to-load test for MonitoringSystemID [ID] was ignored.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Code: ADESTAT-13

Check Name: Determine System Appendix D Status

Related Former Checks:

Applicability:

Description: Determines the Appendix D status for the system based on current system status and the current component accuracy and/or PEI status.

Specifications:

if (***CurrentAppendixDStatus*** == "OOC-Multiple Reasons" OR (***CurrentAppendixDStatus*** starts with "OOC" and ***CurrentAccuracyStatus*** starts with "OOC" and ***CurrentAppendixDStatus*** <> ***CurrentAccuracyStatus***))

Set ***CurrentAppendixDStatus*** = "OOC-Multiple Reasons"

else if (***CurrentAppendixDStatus*** starts with "OOC")

--do nothing

else if (***CurrentAccuracyStatus*** starts with "OOC")

Set ***CurrentAppendixDStatus*** = ***CurrentAccuracyStatus***.

else if ((***CurrentAppendixDStatus*** is not null and does not start with "IC" or "Undetermined" and ***CurrentAppendixDStatus*** does not end with "Not Yet Evaluated") and (***CurrentAccuracyStatus*** does not start with "IC" or "Undetermined" and ***CurrentAccuracyStatus*** does not end with "Not Yet Evaluated") and ***CurrentAppendixDStatus*** <> ***CurrentAccuracyStatus***)

Set ***CurrentAppendixDStatus*** = "Invalid Data".

else if (***CurrentAppendixDStatus*** does not start with "IC" or "Undetermined" and ***CurrentAppendixDStatus*** does not end with "Not Yet Evaluated" AND ***CurrentAppendixDStatus*** is not null)

-- do nothing

else if (***CurrentAccuracyStatus*** does not start with "IC" or "Undetermined" and ***CurrentAccuracyStatus*** does not end with "Not Yet Evaluated")

Set ***CurrentAppendixDStatus*** = ***CurrentAccuracyStatus***.

else if (***CurrentAppendixDStatus*** ends with "Not Yet Evaluated" or ***CurrentAccuracyStatus*** ends with "Not Yet Evaluated")

Set ***CurrentAppendixDStatus*** = "Test Not Yet Evaluated"

else if (***CurrentAppendixDStatus*** starts with "Undetermined" or ***CurrentAccuracyStatus*** starts with "Undetermined")

Set ***CurrentAppendixDStatus*** = "Undetermined"

else if (***CurrentAppendixDStatus*** == "IC-Extension" or ***CurrentAccuracyStatus*** == "IC-Extension")

Set ***CurrentAppendixDStatus*** = "IC-Extension"

else

Set ***CurrentAppendixDStatus*** = "IC"

if (***PEIRequired*** == true)

if (***CurrentAppendixDStatus*** == "OOC-Multiple Reasons" OR (***CurrentAppendixDStatus*** starts with "OOC" and

CurrentPEIStatus starts with "OOC" and ***CurrentAppendixDStatus*** <> ***CurrentPEIStatus***))

Set ***CurrentAppendixDStatus*** = "OOC-Multiple Reasons"

else if (***CurrentAppendixDStatus*** starts with "OOC")

-- do nothing

else if (***CurrentPEIStatus*** starts with "OOC")

Set ***CurrentAppendixDStatus*** = ***CurrentPEIStatus***.

else if ((***CurrentAppendixDStatus*** is not null and does not start with "IC" or "Undetermined" and ***CurrentAppendixDStatus*** does not end with "Not Yet Evaluated") and (***CurrentPEIStatus*** does not start with "IC" or "Undetermined" and ***CurrentPEIStatus*** does not end with "Not Yet Evaluated") and ***CurrentAppendixDStatus*** <> ***CurrentPEIStatus***)

Set ***CurrentAppendixDStatus*** = "Invalid Data".

else if (***CurrentAppendixDStatus*** does not start with "IC" or "Undetermined" and ***CurrentAppendixDStatus*** does not end with "Not Yet Evaluated" AND ***CurrentAppendixDStatus*** is not null)

-- do nothing

else if (***CurrentPEIStatus*** does not start with "IC" or "Undetermined" and ***CurrentPEIStatus*** does not end with "Not Yet Evaluated")

Set ***CurrentAppendixDStatus*** = ***CurrentPEIStatus***.

else if (***CurrentAppendixDStatus*** ends with "Not Yet Evaluated" or ***CurrentPEIStatus*** ends with "Not Yet Evaluated")

Set ***CurrentAppendixDStatus*** = "Test Not Yet Evaluated"

else if (***CurrentAppendixDStatus*** starts with "Undetermined" or ***CurrentPEIStatus*** starts with "Undetermined")

Set ***CurrentAppendixDStatus*** = "Undetermined"

else if (***CurrentAppendixDStatus*** == "IC-Extension" or ***CurrentPEIStatus*** == "IC-Extension")

Set ***CurrentAppendixDStatus*** = "IC-Extension"

else

Set ***CurrentAppendixDStatus*** = "IC"

Results:

Result	Response	Severity

Usage:

1 Process/Category: Emissions Data Evaluation Report ---------- Fuel Flowmeter QA Status Evaluation

Check Category:

Daily Calibration Status

Check Code: DCSTAT-1

Check Name: Locate Most Recent Prior Daily Calibration Test

Related Former Checks:

Applicability: CEM Check

Description: Determines if there is an applicable prior Daily Calibration Test

Specifications:

Set ***PriorDailyCalRecord*** = null.
Set ***PriorDailyCalLastCoveredNonOpHour*** = null.
Set ***PriorDailyCalFirstOpHourAfterLastNonOpHour*** = null.
Set ***InvalidDailyCalRecord*** = null.

Locate the most recent record in ***MostRecentDailyCalibrationTestObject*** for the location where:

a) ComponentID is equal to the ***ApplicableComponentID*** AND
b) ValidFlag is equal to "Y" AND
c) the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed***

if (***MostRecentDailyCalibrationTestObject*** is found)

Set ***PriorDailyCalRecord*** = the DailyCalibrationRecord for the found ***MostRecentDailyCalibrationTestObject***.

if (***QaStatusPrimaryOrPrimaryBypassSystemId*** is null)

Set ***PriorDailyCalLastCoveredNonOpHour*** = the LastCoveredNonOpHour for the found ***MostRecentDailyCalibrationTestObject***.
Set ***PriorDailyCalFirstOpHourAfterLastNonOpHour*** = the FirstOpHourAfterLastCoveredNonOpHour for the found ***MostRecentDailyCalibrationTestObject***.

else if (SystemDictionary for the found ***MostRecentDailyCalibrationTestObject*** contains ***QaStatusPrimaryOrPrimaryBypassSystemId***))

Set ***PriorDailyCalLastCoveredNonOpHour*** = the LastCoveredNonOpHour for the ***QaStatusPrimaryOrPrimaryBypassSystemId*** entry in SystemDictionary for the found ***MostRecentDailyCalibrationTestObject***.
Set ***PriorDailyCalFirstOpHourAfterLastNonOpHour*** = the FirstOpHourAfterLastCoveredNonOpHour for the ***QaStatusPrimaryOrPrimaryBypassSystemId*** entry in SystemDictionary for the found ***MostRecentDailyCalibrationTestObject***.

Locate the most recent record in ***MostRecentDailyCalibrationTestObject*** for the location where:

a) the ComponentID is equal to the ***ApplicableComponentID*** AND
b) ValidFlag is equal to "N" AND
c) the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed***

if (***MostRecentDailyCalibrationTestObject*** is found AND ***MostRecentDailyCalibrationTestObject***.EndDate/Hour/Min is greater than the ***PriorDailyCalRecord***.EndDate/Hour/Min)

Set ***InvalidDailyCalRecord*** = the DailyCalibrationRecord for the found ***MostRecentDailyCalibrationTestObject***.

else

Locate the most recent record in ***MostRecentDailyCalibrationTestObject*** for the location where:

a) the ComponentID is equal to the ***ApplicableComponentID*** AND
b) ValidFlag is equal to "N" AND
c) the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed***

if (***MostRecentDailyCalibrationTestObject*** is found)

Set ***InvalidDailyCalRecord*** = the DailyCalibrationRecord for the found ***MostRecentDailyCalibrationTestObject***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2 Daily Calibration Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- FLOW Daily Calibration Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- Hg Daily Calibration Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- NOX Daily Calibration Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB CO2C, NOXC or O2C Daily Calibr
6	Process/Category:	Emissions Data Evaluation Report --------- O2 Dry Daily Calibration Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- O2 Wet Daily Calibration Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 Daily Calibration Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Daily Calibration Status Evaluation

Check Code: DCSTAT-2

Check Name: Locate Most Recent Prior Event

Related Former Checks:

Applicability: CEM Check

Description: Determines if there is an applicable prior event.

Specifications:

Set ***PriorDailyCalEventRecord*** = null.
Set ***CurrentDailyCalStatus*** = null.

Locate the most recent record in ***QACertificationEventRecords*** where:

a) the ComponentID is equal to the ***ApplicableComponentID*** AND
b) the QACertEventDate/Hour is on or prior to the ***CurrentDateHour*** AND

AND either

a) ***PriorDailyCalRecord*** is null AND the QACertEventDate/Hour is in the ***CurrentReportingPeriod*** OR
b) QACertEventDate/Hour is after the ***PriorDailyCalRecord***.EndDate/Hour

AND either

a) ***DualRangeStatus*** = false OR
b) ***HighRangeComponentID*** <> ***LowRangeComponentID*** OR
c) QACertEventCode <> 20, 25, 26, 30, or 172 and ***CurrentAnalyzerRangeUsed*** = "H" OR
d) QACertEventCode <> 35 or 171 and ***CurrentAnalyzerRangeUsed*** = "L"

if (***QACertificationEventRecords*** is found)

Set ***PriorDailyCalEventRecord*** = the found record in ***QACertificationEventRecords***

If (***PriorDailyCalEventRecord*** is null)

if (***PriorDailyCalRecord*** is null)

if (the number of clock hours between the ***First Day of Operation/First Hour of Operation*** and the ***CurrentDateHour*** is less than 25)

Set ***CurrentDailyCalStatus*** = "IC-Undetermined".

else if (***QaStatusSystemTypeCode*** is equal to "HG" or "HCL", AND the number of clock hours between the ***QaStatusComponentBeginDateHour*** and the ***CurrentDateHour*** is less than 25)

Set ***CurrentDailyCalStatus*** = "IC-Undetermined".

else if (***QaStatusSystemTypeCode*** is equal to "HG" or "HCL", AND ***MatsDailyCalRequiredDate*** is NOT null, AND ***CurrentDateHour*** is before ***MatsDailyCalRequiredDate***)

Set ***CurrentDailyCalStatus*** = "IC-Undetermined".

else (***QaStatusSystemTypeCode*** is equal to "HG" or "HCL", ***QaStatusMatsErbDate*** is not null, AND the number of clock hours between the ***QaStatusMatsErbDate*** hour 0 and the ***CurrentDateHour*** is less than 25)

Set ***CurrentDailyCalStatus*** = "IC-Undetermined".

else if (***QaStatusSystemTypeCode*** is equal to "SO2", ***So2cIsOnlyForMats*** is true, AND ***MatsDailyCalRequiredDate*** is NOT null, AND ***CurrentDateHour*** is before ***MatsDailyCalRequiredDate***)

Set ***CurrentDailyCalStatus*** = "IC-Undetermined".

else

Locate the latest record in ***HourlyOpData*** where the Date/Hour is ON OR PRIOR to the 24th clock hour following the ***First Day of Operation/First Hour of Operation*** and OpTime is equal to zero.

if (***HourlyOpData*** is found)

Locate the first record in ***HourlyOpData*** where the Date/Hour is after the Date/Hour in the ***HourlyOpData*** record found above and ON OR PRIOR to the ***CurrentDateHour*** and the OpTime is greater than zero.

if (not found OR the number of clock hours from ***HourlyOpData***.Date/Hour to the ***CurrentDateHour*** is less than 8)

Set ***CurrentDailyCalStatus*** = "IC-Undetermined".

else

Set ***CurrentDailyCalStatus*** = "OOC-No Prior Test".

else

if (***PriorDailyCalRecord*** .TestResultCode = null)

Set ***CurrentDailyCalStatus*** = "OOC-Test Has Critical Errors".

else if (***PriorDailyCalRecord*** .TestResultCode = "FAILED")

Set ***CurrentDailyCalStatus*** = "OOC-Test Failed".

else if (***PriorDailyCalRecord*** .TestResultCode = "ABORTED")

Set ***CurrentDailyCalStatus*** = "OOC-Test Aborted".

else

Set ***CurrentDailyCalStatus*** = "OOC-Event".

if (***InvalidDailyCalRecord*** is not null and ***InvalidDailyCalRecord***.EndDate/Hour is BEFORE the ***PriorDailyCalEventRecord***.QACertEventDate/Hour)

Set ***InvalidDailyCalRecord*** = null.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2 Daily Calibration Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- FLOW Daily Calibration Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- Hg Daily Calibration Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- NOX Daily Calibration Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB CO2C, NOXC or O2C Daily Calibr
6	Process/Category:	Emissions Data Evaluation Report --------- O2 Dry Daily Calibration Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- O2 Wet Daily Calibration Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 Daily Calibration Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Daily Calibration Status Evaluation

Check Code: DCSTAT-3

Check Name: Determine Test Expiration Date for Most Recent Prior Daily Calibration Test

Related Former Checks:

Applicability: CEM Check

Description: Determines the expiration dates for the Applicable Prior Daily Calibration test.

Specifications:

Set ***OnlineDailyCalRecord*** to null.

if (***CurrentDailyCalStatus*** is null)

if (***PriorDailyCalRecord***.OnlineIndicator == 1)

if (the number of clock hours between the ***PriorDailyCalRecord***.Date/Hour and the ***CurrentDateHour*** is less than 26)

Set ***CurrentDailyCalStatus*** = "IC".

else

if (***PriorDailyCalLastCoveredNonOpHour*** is NOT null)

if (***PriorDailyCalFirstOpHourAfterLastNonOpHour*** is NOT null) AND (the number of clock hours between the ***PriorDailyCalFirstOpHourAfterLastNonOpHour*** and the ***CurrentDateHour*** is greater than or equal to 8)

Set ***CurrentDailyCalStatus*** = "OOC-Expired".

else

Set ***CurrentDailyCalStatus*** = "IC-Grace".

else

Set ***CurrentDailyCalStatus*** = "OOC-Expired".

else

Locate the most recent record in ***MostRecentDailyCalibrationTestObject*** for the location where:

a) ComponentID is equal to the ***ApplicableComponentID*** AND
b) ValidFlag is equal to "Y" AND
c) the OnlineIndicator = 1 AND
d) the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed***

if (***MostRecentDailyCalibrationTestObject*** is found)

Set ***OnlineDailyCalRecord*** = the DailyCalibrationRecord for the found ***MostRecentDailyCalibrationTestObject***.

if (***QaStatusPrimaryOrPrimaryBypassSystemId*** is null)

Set *OnlineDailyCalOpHourCount* = the OperatingHourCount for the found ***MostRecentDailyCalibrationTestObject***.
Set *OnlineDailyCalLastCoveredNonOpHour* = the LastCoveredNonOpHour for the found ***MostRecentDailyCalibrationTestObject***.
Set *OnlineDailyCalFirstOpHourAfterLastNonOpHour* = the FirstOpHourAfterLastCoveredNonOpHour

for the found ***MostRecentDailyCalibrationTestObject***.

else if (SystemDictionary for the found ***MostRecentDailyCalibrationTestObject*** contains ***QaStatusPrimaryOrPrimaryBypassSystemId***))

Set *OnlineDailyCalOpHourCount* = the OperatingHourCount for the ***QaStatusPrimaryOrPrimaryBypassSystemId*** entry in SystemDictionary for the found ***MostRecentDailyCalibrationTestObject***.
Set *OnlineDailyCalLastCoveredNonOpHour* = the LastCoveredNonOpHour for the ***QaStatusPrimaryOrPrimaryBypassSystemId*** entry in SystemDictionary for the found ***MostRecentDailyCalibrationTestObject***.
Set *OnlineDailyCalFirstOpHourAfterLastNonOpHour* = the FirstOpHourAfterLastCoveredNonOpHour for the ***QaStatusPrimaryOrPrimaryBypassSystemId*** entry in SystemDictionary for the found ***MostRecentDailyCalibrationTestObject***.

else

Set *OnlineDailyCalOpHourCount* = null.
Set *OnlineDailyCalLastCoveredNonOpHour* = null.
Set *OnlineDailyCalFirstOpHourAfterLastNonOpHour* = null.

if (***InvalidDailyCalRecord*** is null)

Locate the record in ***MostRecentDailyCalibrationTestObject*** for the location where:

a) the ComponentID is equal to the ***ApplicableComponentID*** AND
b) ValidFlag is equal to "N" AND
c) the OnlineIndicator = 1 AND
d) the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed***

if (***MostRecentDailyCalibrationTestObject*** is found AND the ***MostRecentDailyCalibrationTestObject***.EndDate/Hour is after the *OnlineDailyCalRecord***.**EndDate/Hour AND is equal to or prior to the ***PriorDailyCalRecord***.Date/Hour)

set ***InvalidDailyCalRecord*** = the DailyCalibrationRecord for the found ***MostRecentDailyCalibrationTestObject***.

if (***OnlineDailyCalRecord*** .TestResultCode = null)

Set ***CurrentDailyCalStatus*** = "OOC-Prior Online Test Has Critical Errors".

else if (***OnlineDailyCalRecord*** .TestResultCode = "FAILED")

Set ***CurrentDailyCalStatus*** = "OOC-Prior Online Test Failed".

else if (***OnlineDailyCalRecord***.TestResultCode = "ABORTED")

Set ***CurrentDailyCalStatus*** = "OOC-Prior Online Test Aborted".

else if (*OnlineDailyCalOpHourCount* is not null AND *OnlineDailyCalOpHourCount* is less than or equal to 26 AND the number of clock hours between the ***PriorDailyCalRecord***.Date/Hour and the ***CurrentDateHour*** is less than 26)

Set ***CurrentDailyCalStatus*** = "IC".

else if (the number of clock hours between the ***OnlineDailyCalRecord***.Date/Hour and the ***CurrentDateHour*** is

less than 26)

Set ***CurrentDailyCalStatus*** = "IC".

else

if (*OnlineDailyCalLastCoveredNonOpHour* is NOT null)

if (*OnlineDailyCalFirstOpHourAfterLastNonOpHour* is NOT null) AND (the number of clock hours between the *OnlineDailyCalFirstOpHourAfterLastNonOpHour* and the ***CurrentDateHour*** is greater than or equal to 8)

Set ***CurrentDailyCalStatus*** = "OOC-Expired".

else

Set ***CurrentDailyCalStatus*** = "IC-Grace".

else

Set ***CurrentDailyCalStatus*** = "OOC-Expired".

else

if (***Rpt Period Op Hour Accumulator Array*** for the location is less than 26)

Set ***CurrentDailyCalStatus*** = "IC-Undetermined".

else

Set ***CurrentDailyCalStatus*** = "OOC-Expired".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2 Daily Calibration Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- FLOW Daily Calibration Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- Hg Daily Calibration Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- NOX Daily Calibration Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB CO2C, NOXC or O2C Daily Calibr
6	Process/Category:	Emissions Data Evaluation Report --------- O2 Dry Daily Calibration Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- O2 Wet Daily Calibration Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 Daily Calibration Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Daily Calibration Status Evaluation

Check Code: DCSTAT-4

Check Name: Determine Final Daily Calibration Status

Related Former Checks:

Applicability: CEM Check

Description: Evaluates the determined Daily Calibration Status and changes it if needed based on an ignored test or the status of the alternate range.

Specifications:

Set ***AlternateDailyCalRecord*** = null.

if (***CurrentDailyCalStatus*** begins with "OOC")

    if (***InvalidDailyCalRecord*** is not null)

        Set ***CurrentDailyCalStatus*** = ***CurrentDailyCalStatus*** & "*".

    Return result ***CurrentDailyCalStatus***.

else if (***DualRangeStatus*** = true and ***CurrentDailyCalStatus*** begins with "IC")

    if (***CurrentAnalyzerRangeUsed*** = "H")

        Set *AlternateAnalyzerRange* = "L".
        Set *AlternateComponentID* = ***LowRangeComponentID***.

    else

        Set *AlternateAnalyzerRange* = "H".
        Set *AlternateComponentID* = ***HighRangeComponentID***.

    Locate the most recent record in ***DailyCalTestRecordsByLocationForQAStatus*** for the location where:

    a) ComponentID is equal to the *AlternateComponentID* AND
    b) ValidFlag = "Y" AND
    c) SpanScaleCode is equal to the *AlternateAnalyzerRange*

    if (***DailyCalTestRecordsByLocationForQAStatus*** is found)

        Set ***AlternateDailyCalRecord*** = the found record in ***DailyCalTestRecordsByLocationForQAStatus***.

    if (***AlternateDailyCalRecord*** is not null)

        if (***AlternateDailyCalRecord***.TestResultCode = null)

            Set ***CurrentDailyCalStatus*** = "OOC-Alternate Range Test Has Critical Errors".

        else if (***AlternateDailyCalRecord.***TestResultCode = "FAILED")

            Set ***CurrentDailyCalStatus*** = "OOC-Alternate Range Test Failed".

        else if (***AlternateDailyCalRecord***.TestResultCode = "ABORTED")

            Set ***CurrentDailyCalStatus*** = "OOC-Alternate Range Test Aborted".

        else

Locate the latest record in ***DailyCalTestRecordsByLocationForQAStatus*** for the location where:

a) ComponentID is equal to the *AlternateComponentID* AND
b) the SpanScaleCode is equal to the *AlternateAnalyzerRange*
c) the TestResultCode is equal to "FAILED" or "ABORTED"

if (***DailyCalTestRecordsByLocationForQAStatus*** is found AND (either the ***PriorDailyCalRecord*** is null or EndDate/Hour/Minute is after the ***PriorDailyCalRecord***.EndDate/Hour/Minute))

Set ***CurrentDailyCalStatus*** = "OOC-No Passing Test After Alternate Range Failed Test".
(Report this status in the Evaluation Report under the ***PriorDailyCalRecord***.TestDate/Hour.)

Else

Locate the latest record in ***DailyCalTestRecordsByLocationForQAStatus*** for the location where:

a) ComponentID is equal to the ***ApplicableComponentID*** AND
b) the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed***
c) the TestResultCode is equal to "FAILED" or "ABORTED"

if (***DailyCalTestRecordsByLocationForQAStatus*** is found AND EndDate/Hour/Minute is after the ***AlternateDailyCalRecord***.EndDate/Hour/Minute)

Set ***CurrentDailyCalStatus*** = "OOC-No Passing Alternate Range Test After Failed Test".
(Report this status in the Evaluation Report under the ***PriorDailyCalRecord***.TestDate/Hour.)

if (***CurrentDailyCalStatus*** begins with "OOC")

if (***InvalidDailyCalRecord*** is not null)

Set ***CurrentDailyCalStatus*** = ***CurrentDailyCalStatus*** & "*".

else

Locate the most recent record in ***DailyCalTestRecordsByLocationForQAStatus*** for the location where:

a) ComponentID is equal to the *AlternateComponentID* AND
b) ValidFlag is equal to "N" AND
c) the SpanScaleCode is equal to the *AlternateAnalyzerRange*

if (***DailyCalTestRecordsByLocationForQAStatus*** is found AND the EndDate/Hour is after the ***AlternateDailyCalRecord***.EndDate/Hour)

Set ***InvalidDailyCalRecord*** = the found record in ***DailyCalTestRecordsByLocationForQAStatus.***
Set ***CurrentDailyCalStatus*** = ***CurrentDailyCalStatus*** & "*".

Return result ***CurrentDailyCalStatus*** .

elseif (***CurrentDailyCalStatus*** does not begin with "IC")

Return result ***CurrentDailyCalStatus*** .

Results:

Result	Response	Severity
OOC-Alternate Range Test Aborted	The prior daily calibration test for the alternate range [altscale] of [compkey], which was completed on [altdate], was aborted.	Critical Error Level 1
OOC-Alternate Range Test Aborted*	The prior daily calibration test for the alternate range [altscale] of [compkey], which was completed on [altdate], was aborted. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-Alternate Range Test Failed	The prior daily calibration test for the alternate range [altscale] of [compkey], which was completed on [altdate], failed.	Critical Error Level 1
OOC-Alternate Range Test Failed*	The prior daily calibration test for the alternate range [altscale] of [compkey], which was completed on [altdate], failed. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-Alternate Range Test Has Critical Errors	The prior daily calibration test for the alternate range [altscale] of [compkey], which was completed on [altdate], has critical errors.	Critical Error Level 1
OOC-Alternate Range Test Has Critical Errors*	The prior daily calibration test for the alternate range [altscale] of [compkey], which was completed on [altdate], has critical errors. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-Event	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for [compkey], but you did not perform a subsequent daily calibration test.	Critical Error Level 1
OOC-Event*	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for [compkey], but you did not perform a subsequent daily calibration test. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-Expired	The prior daily calibration test for [compkey] completed on [date] has expired.	Critical Error Level 1
OOC-Expired*	The prior daily calibration test for [compkey] completed on [date] has expired. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-No Passing Alternate Range Test After Failed Test	The prior daily calibration test for [compkey] was completed on [date], however a subsequent passing test on [altscale] has not been completed. When a daily calibration test is failed for a dual-range analyzer, you must complete a passing daily calibration test on both ranges before the monitor is considered to be in-control. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-No Passing Alternate Range Test After Failed Test*	The prior daily calibration test for [compkey] was completed on [date], however a subsequent passing test on [altscale] has not been completed. When a daily calibration test is failed for a dual-range analyzer, you must complete a passing daily calibration test on both ranges before the monitor is considered to be in-control. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-No Passing Test After Alternate Range Failed Test	The prior daily calibration test for [compkey] was completed on [date], which is prior to a failed or aborted test for the alternate range [altscale]. When a daily calibration test is failed for a dual-range analyzer, you must complete a passing daily calibration test on both ranges before the monitor is considered to be in-control.	Critical Error Level 1
OOC-No Passing Test After Alternate Range Failed Test*	The prior daily calibration test for [compkey] was completed on [date], which is prior to a failed or aborted test for the alternate range [altscale]. When a daily calibration test is failed for a dual-range analyzer, you must complete a passing daily calibration test on both ranges before the monitor is considered to be in-control. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-No Prior Test	You did not report a prior daily calibration test for [compkey] during the reporting period. Any daily calibration test that may have been completed in a prior reporting period has expired.	Critical Error Level 1
OOC-No Prior Test*	You did not report a prior daily calibration test for [compkey] during the reporting period. Any daily calibration test that may have been completed in a prior reporting period has expired. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-No Probationary Calibration Test	This check result is obsolete.	No Errors

OOC-No Probationary Calibration Test*	This check result is obsolete.	No Errors
OOC-Prior Online Test Aborted	The prior online daily calibration test for [compkey] completed on [ondate] was aborted.	Critical Error Level 1
OOC-Prior Online Test Aborted*	The prior online daily calibration test for [compkey] completed on [ondate] was aborted. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-Prior Online Test Expired	The prior online daily calibration test for [compkey] completed on [ondate] has expired.	Critical Error Level 1
OOC-Prior Online Test Expired*	The prior online daily calibration test for [compkey] completed on [ondate] has expired. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-Prior Online Test Failed	The prior online daily calibration test for [compkey] completed on [ondate] failed.	Critical Error Level 1
OOC-Prior Online Test Failed*	The prior online daily calibration test for [compkey] completed on [ondate] failed. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-Prior Online Test Has Critical Errors	The prior online daily calibration test for [compkey] completed on [ondate] has critical errors.	Critical Error Level 1
OOC-Prior Online Test Has Critical Errors*	The prior online daily calibration test for [compkey] completed on [ondate] has critical errors. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-Test Aborted	The prior daily calibration test for [compkey] completed on [date] was aborted.	Critical Error Level 1
OOC-Test Aborted*	The prior daily calibration test for [compkey] completed on [date] was aborted. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-Test Failed	The prior daily calibration test for [compkey] completed on [date] failed.	Critical Error Level 1
OOC-Test Failed*	The prior daily calibration test for [compkey] completed on [date] failed. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1
OOC-Test Has Critical Errors	The prior daily calibration test for [compkey] completed on [date] has critical errors.	Critical Error Level 1
OOC-Test Has Critical Errors*	The prior daily calibration test for [compkey] completed on [date] has critical errors. An invalid daily calibration test completed on [invdate] was ignored.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2 Daily Calibration Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- FLOW Daily Calibration Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- Hg Daily Calibration Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- NOX Daily Calibration Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB CO2C, NOXC or O2C Daily Calibr
6	Process/Category:	Emissions Data Evaluation Report --------- O2 Dry Daily Calibration Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- O2 Wet Daily Calibration Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 Daily Calibration Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Daily Calibration Status Evaluation

Check Category:

Daily Calibration Test

Check Code: DAYCAL-1

Check Name: Daily Calibration Test Component Type Check

Related Former Checks:

Applicability: CEM Check

Description: This check determines whether the component type reported is appropriate for an Daily Calibration test.

Specifications:

For the daily calibration test:

Set ***Daily Cal Calc Result*** to null.
Set ***Daily Cal Fail Date*** and ***Daily Cal Fail Hour*** to null.

If the ComponentID is null,

set ***Daily Cal Component Type Valid*** to false.
return result A.

Otherwise,

If the ComponentTypeCode of the associated component is equal to "SO2", "NOX", "CO2", "O2", or "FLOW",

set ***Daily Cal Component Type Valid*** to true.

Else if the ComponentTypeCode of the associated component is equal to "HG" or "HCL",

If (OnlineOfflineIndicator is equal to 1)

set ***Daily Cal Component Type Valid*** to true.

Else
set ***Daily Cal Component Type Valid*** to false.
return result C.

Otherwise,

set ***Daily Cal Component Type Valid*** to false.
return result B.

If component is invalid, do not perform injection-based checks. Set the calculated values to null.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Fatal
B	The ComponentTypeCode in the monitoring plan is [comptype]. This type of component does not require a calibration test. Only component types 'SO2', 'NOX', 'CO2', 'O2', "HG", or 'FLOW' may have a daily calibration test.	Critical Error Level 1
C	For Hg and or HCl CEMS, all calibrations must be done while unit is online.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-2

Check Name: Aborted and Incomplete Daily Calibration Test Check

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

For the daily calibration test:

Set ***Evaluate Upscale Injection*** AND ***Evaluate Zero Injection*** to false.

If ***Daily Cal Component Type Valid*** is equal to true,

If the TestResultCode is equal to "ABORTED",
set ***Daily Cal Calc Result*** to "ABORTED", and return result A.

If the TestResultCode is equal to "INC",
set ***Daily Cal Calc Result*** to "INC".

If ZeroInjectionDate, ZeroInjectionHour, and ZeroMeasuredValue are not null,
set ***Evaluate Zero Injection*** to true.

If UpscaleInjectionDate, UpscaleInjectionHour, and UpscaleMeasuredValue are not null,
set ***Evaluate Upscale Injection*** to true.

Otherwise,
set ***Evaluate Upscale Injection*** AND ***Evaluate Zero Injection*** to true.

Results:

Result	Response	Severity
A	The TestResultCode indicates that the [type] test for [key] was aborted. If the test was aborted for a reason not related to monitor performance, you should not report the test.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-3

Check Name: Online Offline Indicator Valid

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

For a daily calibration test:

Set ***Daily Cal Calc Online Ind*** to null.

If ZeroOpTime is equal to 0 OR UpscaleOpTime is equal to 0
set ***Daily Cal Calc Online Ind*** to 0

else if ZeroOpTime is equal to 1 AND UpscaleOpTime is equal to 1
set ***Daily Cal Calc Online Ind*** to 1

else
set ***Daily Cal Calc Online Ind*** to the OnlineOfflineIndicator.

If (OnlineOfflineIndicator is null)
return result A.

else if (ComponentTypeCode is equal to "HG" or "HCL")

If (OnlineOfflineIndicator is equal to 1) // Earlier component type error will occur if indicator is not 1

If (***Daily Cal Calc Online Ind*** is equal to 0)
return result E.

else

If (***Daily Cal Calc Online Ind*** is equal to 0)

If (OnlineOfflineIndicator is equal to 1)
return result B.

else

Locate the latest ***OOC Test Record*** for the location where the ComponentID and SpanScaleCode is equal to the ComponentID and SpanScaleCode in the current test and the EndDate/Hour is prior to the Date/Hour of the current test.

If not found,
Set ***Ignored Daily Calibration Tests*** to true.
If (***Daily Cal Calc Result*** <> "INVALID")
set ***Daily Cal Calc Result*** to "IGNORED"

Otherwise,
Locate an ***QA Certification Event Record*** for the location where the ComponentID is equal to the ComponentID in the current test AND OOCRequired == "Y" AND the EventDate/Hour is after the EndDate/Hour of the retrieved OOC test AND the EventDate/Hour is on or before the EndDate/Hour of the current test AND EITHER

a) SpanScaleCode in the current test is null OR
b) SpanScaleCode in the current test == "H" and QACertEventCode <> 20, 25, 26, 30, or 172

OR
c) SpanScaleCode in the current test == "L" and QACertEventCode <> 35 or 171

If found,
return result D.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	The OnlineOfflineIndicator in the daily calibration test indicates that the test was performed on-line, but OperatingTime in the Hourly Operating Data record is 0.	Critical Error Level 1
C	This check result is obsolete.	No Errors
D	The test was peformed while the unit was not operating, but this is not valid, because you reported an QA Certification Event record indicating that you needed to perform an online-offline calibration demonstration allowing you to conduct off-line daily calibration tests. However, you have not reported an online-offline calibration demonstration subsequent to the EventDate and EventHour in the QA Certification Event record.	Critical Error Level 2
E	For Hg and or HCl CEMS, all calibrations must be done while unit is online.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-4

Check Name: Test Span Scale Valid

Related Former Checks:

Applicability: CEM Check

Description: This check determines whether the reported span scale is valid and consistent with the current analyzer range of the component.

Specifications:

For a daily calibration test with a valid component:

Set ***Daily Cal Span Scale Valid*** to true.

If the ComponentTypeCode of the associated component is not equal to "FLOW", not equal to "HG", or not equal to "HCL",

If the SpanScaleCode is null,
set ***Daily Cal Span Scale Valid*** to false, and return result A.

If the SpanScaleCode is not equal to "H" or "L",
set ***Daily Cal Span Scale Valid*** to false, and return result B.

If the ***EM Test Date Valid*** AND ***EM Test Hour Valid*** are true,

If the SpanScaleCode is equal to "H"

Locate an Analyzer Range record for the component where the AnalyzerRangeCode is equal to "L", the BeginDate and BeginHour is on or before the Date and Hour in the current test, and the EndDate is null or the EndDate and EndHour is on or after the Date and Hour of the current test.

If found,
set ***Daily Cal Span Scale Valid*** to false, and return result C.

If the SpanScaleCode is equal to "L"

Locate an Analyzer Range record for the component where the AnalyzerRangeCode is equal to "H", the BeginDate and BeginHour is on or before the Date and Hour of the current test, and the EndDate is null or the EndDate and EndHour is on or after the Date and Hour of the current test.

If found,
set ***Daily Cal Span Scale Valid*** to false, and return result C.

Else, if the ComponentTypeCode of the associated component is equal to "HG" or "HCL",

If the SpanScaleCode is null,
set ***Daily Cal Span Scale Valid*** to false, and return result A.

Else if the SpanScaleCode is not equal to "H",
set ***Daily Cal Span Scale Valid*** to false, and return result B.

Otherwise,
If the SpanScaleCode is not null,
set ***Daily Cal Span Scale Valid*** to false, and return result D.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	You reported the value [value], which is not in the list of valid values, in the field [fieldname] for [key].	Critical Error Level 1
C	The active analyzer range for the component is inconsistent with the span scale [value] reported for the [type] test for [key].	Critical Error Level 1
D	You reported a SpanScaleCode in the [type] test for [key], but this is not appropriate for flow component.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-5

Check Name: Determine Span Value

Related Former Checks:

Applicability: CEM Check

Description: This check determines the span value for the test:

Specifications:

For a daily calibration test:

Set ***Daily Cal Span Value*** to null.

If ***EM Test Date Valid***, ***EM Test Hour Valid***, and ***Daily Cal Span Scale Value*** are all true,

Locate the System Component records for the associated component with the earliest Begin Date.

If found,

If the BeginDate in the retrieved record is not null, the BeginHour in the retrieved record is between 0 and 23, and the BeginDate and BeginHour is later than the Date and Hour of the test.

Locate a Span Record for the location where the ComponentTypeCode equal to the ComponentTypeCode of the associated component, the SpanScaleCode is equal to the SpanScaleCode in the test, the Span Value is greater than 0, the BeginDate and BeginHour is on or before the BeginDate and BeginHour of the retrieved record, and the EndDate is null or the EndDate and EndHour is after the BeginDate and BeginHour of the retrieved record.

Otherwise,

Locate a Span Record for the location where the ComponentTypeCode equal to the ComponentTypeCode of the associated component, the SpanScaleCode is equal to the SpanScaleCode in the test, the Span Value is greater than 0, the BeginDate and BeginHour is on or before the Date and Hour of the test, and the EndDate is null or the EndDate and EndHour is after the Date and Hour of the test.

If not found,
return result A.

If more than one record is found,
return result B.

If one record is found,
set ***Daily Cal Span Value*** to the SpanValue in the retrieved span record.

else
return result C.

Results:

Result	Response	Severity
A	You have not reported a valid monitoring plan span record that was active during the test for [key].	Critical Error Level 1
B	You reported more than one monitoring plan span record that was active during the test for [key].	Critical Error Level 1
C	The component reported for [key] is not part of any monitoring system.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-6

Check Name: Daily Calibration Test Upscale Gas Level Code Valid

Related Former Checks:

Applicability: CEM Check

Description: This check is to make sure that the Upscale Gas Level Code is valid.

Specifications:

For the daily calibration test with an upscale injection:

If the UpscaleGasCode is null,
set ***Daily Cal Upscale Gas Level Valid*** to false, and return result A.

If the UpscaleGasCode is not equal to "MID" or "HIGH",
set ***Daily Cal Upscale Gas Level Valid*** to false, and return result B.

If the ComponentTypeCode of the associated component is equal to "FLOW", and the UpscaleGasLevelCode is equal to "MID",
set ***Daily Cal Upscale Gas Level Valid*** to false, and return result C.

Otherwise,
set ***Daily Cal Upscale Gas Level Valid*** to true.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	You reported the value [value], which is not in the list of valid values, in the field [fieldname] for [key].	Critical Error Level 1
C	You have reported a value of "MID" as the UpscaleGasCode. This value is not appropriate for flow components.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-7

Check Name: Reference Values Consistent with Calibration Gas Levels

Related Former Checks:

Applicability: CEM Check

Description: This check is to identify reference values which are not correct relative to the calibration gas levels indicated.

Specifications:

For the daily calibration test with an upscale and zero injection:

If ZeroReferenceValue greater than or equal to 0, UpscaleReferenceValue greater than 0, AND ZeroReferenceValue is greater than or equal to UpscaleReferenceValue,
set ***Daily Cal Calc Result*** to "INVALID", and return result A.

Results:

Result	Response	Severity
A	The reference value is not consistent with the reported calibration gas levels in the daily calibration test for [key]. The reference values of zero-level gas injection or reference signals must be less than that of the upscale gas injection.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-8

Check Name: Zero Measured Value Valid

Related Former Checks:

Applicability: CEM Check

Description: This check is to make sure that the Zero Measured Value is reported.

Specifications:

For the daily calibration test with a zero injection:

If ZeroMeasuredValue is null,
return result A.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-9

Check Name: Zero Reference Value Valid

Related Former Checks:

Applicability: CEM Check

Description: This check is to make sure that the Zero Reference Value is reported.

Specifications:

For the daily calibration test with a zero injection:

If ZeroReferenceValue is null,
return result A.

If ZeroReferenceValue is less than 0,
return result B.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	The value [value] in the field [fieldname] for [key] is not within the range of valid values. This value must be greater than or equal to zero.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-10

Check Name: Zero Calibration Error Valid

Related Former Checks:

Applicability: CEM Check

Description: This check is to make sure that the Zero Calibration Error is reported.

Specifications:

For the daily calibration test with a zero injection:

If the ZeroCalibrationError is null,
return result A.

If the ZeroCalibrationError is less than 0,
return result B.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	The value [value] in the field [fieldname] for [key] is not within the range of valid values. This value must be greater than or equal to zero.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-11

Check Name: Zero APS Indicator Valid

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

For the daily calibration test with a zero injection:

If ZeroAPSIndicator is null,
return result A.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-12

Check Name: Upscale Measured Value Valid

Related Former Checks:

Applicability: CEM Check

Description: This check is to make sure that the Upscale Measured Value is reported.

Specifications:

For the daily calibration test with an upscale injection:

If UpscaleMeasuredValue is null,
return result A.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-13

Check Name: Upscale Reference Value Valid

Related Former Checks:

Applicability: CEM Check

Description: This check is to make sure that the Upscale Reference Value is reported.

Specifications:

For the daily calibration test with an upscale injection:

If UpscaleReferenceValue is null,
return result A.

If UpscaleReferenceValue is less than or equal to 0,
return result B.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	The value [value] in the field [fieldname] for [key] is not within the range of valid values. This value must be greater than zero.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-14

Check Name: Upscale Calibration Error Valid

Related Former Checks:

Applicability: CEM Check

Description: This check is to make sure that the Upscale Calibration Error is reported.

Specifications:

For the daily calibration test with an upscale injection:

If the UpscaleCalibrationError is null,
return result A.

If the UpscaleCalibrationError is less than 0,
return result B.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	The value [value] in the field [fieldname] for [key] is not within the range of valid values. This value must be greater than or equal to zero.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-15

Check Name: Upscale APS Indicator Valid

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

For the daily calibration test with an upscale injection:

If UpscaleAPSIndicator is null,
return result A.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-16

Check Name: Upscale Injection Time Valid

Related Former Checks:

Applicability: CEM Check

Description: This check determines whether the Injection Date and Hour reported in the Injection Element is valid.

Specifications:

For the daily calibration test with an upscale injection:

If the UpscaleInjectionHour is not between 0 and 23, or the UpscaleInjectionMinute is null and ***Legacy Data Evaluation*** == false, or the UpscaleInjectionMinute is not between 0 and 59,

set ***Daily Cal Upscale Injection Time Valid*** to false, and return result A.

Otherwise,
set ***Daily Cal Upscale Injection Time Valid*** to true.

Results:

Result	Response	Severity
A	The [type] date, hour, and/or minute for [key] is invalid.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-17

Check Name: Zero Injection Time Valid

Related Former Checks:

Applicability: CEM Check

Description: This check determines whether the Injection Date and Hour reported in the Injection Element is valid.

Specifications:

For the daily calibration test with a zero injection:

If the ZeroInjectionHour is not between 0 and 23, or the ZeroInjectionMinute is null and ***Legacy Data Evaluation*** == false, or the ZeroInjectionMinute is not between 0 and 59,

set ***Daily Cal Injection Times Valid*** to false, and return result A.

else if ***Legacy Data Evaluation*** == false, the UpscaleInjectionDate is not null, the UpscaleInjectionHour is between 0 and 23, the UpscaleInjectionMinute is between 0 and 59, and the UpscaleInjectionDate, UpscaleInjectionHour, and UpscaleInjectionMinute are equal to the ZeroInjectionDate, ZeroInjectionHour, and ZeroInjectionMinute, and the associated ComponentTypeCode is not equal to "FLOW"

set ***Daily Cal Injection Times Valid*** to false, and return result B.

Otherwise,
set ***Daily Cal Injection Times Valid*** to ***Daily Cal Upscale Injection Time Valid***.

Locate another ***Daily Calibration Test Record*** for the location where the ComponentID and SpanScale are equal to the ComponentID and SpanScale in the current record, TestResultCode is not equal to "INC", and the EndDate/Hour/Minute is between the UpscaleInjectionDate/Hour/Minute and ZeroInjectionDate/Hour/Minute of the current test.

If found,
return result C.

else
If the absolute value of the difference between the ZeroInjectionDate/Hour and the UpscaleInjectionDate/Hour in the current test is greater than 1,
return result D.

Results:

Result	Response	Severity
A	The [type] date, hour, and/or minute for [key] is invalid.	Critical Error Level 1
B	You reported that the zero injection and upscale injection for [key] were performed at the same time. This is invalid.	Critical Error Level 1
C	This [testtype] was conducted at the same time as another [testtype] for the same component and range.	Critical Error Level 1
D	The zero and upscale injections for [key] were not performed in the same or adjacent clock hours.	Critical Error Level 2

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-18

Check Name: Zero Reference Value Consistent with Span

Related Former Checks:

Applicability: CEM Check

Description: This check determines whether the calibration gas or reference signal is appropriate for span and gas level.

Validation Tables:

Test Tolerances (Cross Check Table)

Specifications:

For the daily calibration test with a ***Daily Cal Span Value*** that is not null and a ZeroReferenceValue greater than or equal to 0:

If the ComponentTypeCode of the associated component is not equal to "HG",

Calculate ***Zero Reference Percent of Span*** = ZeroReferenceValue / ***Daily Cal Span Value*** * 100, and round to result to one decimal place.

If ***Zero Reference Percent of Span*** is greater than 20.0,

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "GasPercentOfSpan".

If ***Zero Reference Percent of Span*** is greater than 20.0 + Tolerance in the cross-check record,
return result A.

Otherwise,
return result B.

Results:

Result	Response	Severity
A	The tag value of at least one Zero level reference signal or calibration gas for [key] is [percent]%, which does not meet the performance specifications of 40 CFR Part 75. The concentration of the zero reference signal or calibration gas must be less than or equal to 20.0% of the span value. The test is invalid.	Critical Error Level 2
B	The tag value of at least one zero level reference signal or calibration gas for [key] is [percent]%, which does not meet the performance specifications of 40 CFR Part 75. The concentration of the zero reference signal or calibration gas must be less than or equal to 20.0% of the span value.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-19

Check Name: Upscale Reference Value Consistent with Span

Related Former Checks:

Applicability: CEM Check

Description: This check determines whether the calibration gas or reference signal is appropriate for span and gas level.

Validation Tables:

Test Tolerances (Cross Check Table)

Specifications:

For the daily calibration test with a ***Daily Cal Span Value*** that is not null and an UpscaleReferenceValue greater than 0:

Calculate ***Upscale Reference Percent of Span*** = UpscaleReferenceValue / ***Daily Cal Span Value*** * 100, and round to result to one decimal place.

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "GasPercentOfSpan".

If UpscaleGasLevelCode is equal to "MID", and the ComponentTypeCode of the associated component is not equal to "FLOW",

If ***Upscale Reference Percent of Span*** is less than 50.0 or greater than 60.0,

If ***Upscale Reference Percent of Span*** is less than 50.0 - Tolerance in the cross-check record or ***Upscale Reference Percent of Span*** greater than 60.0 + Tolerance in the cross-check record,
return result A.

Otherwise,
return result B.

If UpscaleGasLevelCode is equal to "HIGH",

If the ComponentTypeCode of the associated component is equal to "FLOW",

If ***Upscale Reference Percent of Span*** is less than 50.0 or greater than 70.0,

If ***Upscale Reference Percent of Span*** is less than 50.0 - Tolerance in the cross-check record or ***Upscale Reference Percent of Span*** greater than 70.0 + Tolerance in the cross-check record,
return result C.

Otherwise,
return result D.

Otherwise,

If ***Upscale Reference Percent of Span*** is greater than 100.0,
return result E

If ***Upscale Reference Percent of Span*** is less than 80.0,

If ***Upscale Reference Percent of Span*** is less than 80.0 - Tolerance in the cross-check record,
return result E.

Otherwise,
return result F.

Results:

Result	Response	Severity
A	The tag value of at least one Mid level reference signal or calibration gas for [key] is [percent]%, which does not meet the applicable performance specifications. The concentration of the mid reference signal or calibration gas must be between 50.0% and 60.0% of the span value. The test is invalid.	Critical Error Level 2
B	The tag value of at least one Mid level reference signal or calibration gas for [key] is [percent]%, which does not meet the applicable performance specifications. The concentration of the 'mid' reference signal or calibration gas must be between 50.0% and 60.0% of the span value.	Non-Critical Error
C	The tag value of at least one High level reference signal for [key] is [percent]%, which does not meet the performance specifications of 40 CFR Part 75. The value of the high reference signal for a flow component must be between 50.0% and 70.0% of the span value. The test is invalid.	Critical Error Level 2
D	The tag value of at least one High level reference signal for [key] is [percent]%, which does not meet the performance specifications of 40 CFR Part 75. The value of the 'high' reference signal for a flow component must be between 50.0% and 70.0% of the span value.	Non-Critical Error
E	The tag value of at least one High level reference calibration gas for [key] is [percent]%, which does not meet the applicable performance specifications. The concentration of the high reference calibration gas must be between 80.0% and 100.0% of the span value. The test is invalid.	Critical Error Level 2
F	The tag value of at least one High level reference calibration gas for [key] is [percent]%, which does not meet the applicable performance specifications. The concentration of the 'high' reference calibration gas must be between 80.0% and 100.0% of the span value.	Non-Critical Error

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-20

Check Name: Calculate Zero Gas Injection or Reference Signal Results

Related Former Checks:

Applicability: CEM Check

Description: This check is to calculate calibration errors.

Validation Tables:

Test Tolerances (Cross Check Table)

Specifications:

For the daily calibration test with a zero injection:

If (***Daily Cal Span Value*** is null, or ZeroReferenceValue of the test is null or is less than zero, or ZeroMeasuredValue of the test is null)

Set ***Daily Cal Calc Result*** to "INVALID", ***Daily Cal Zero Injection Calc Result*** to null, ***Daily Cal Zero Injection Calc APS Indicator*** to null, and return result A.

Otherwise,

Calculate *diff* = abs(ZeroMeasuredValue - ZeroReferenceValue)
Set ***Daily Cal Zero Injection Calc APS Indicator*** to 0.

If (ComponentTypeCode of the associated component is equal to "CO2" or "O2")

Round *diff* to 1 decimal place.
Set ***Daily Cal Zero Injection Calc Result*** to *diff*.

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "IGNORED")

If (***Daily Cal Zero Injection Calc Result*** is greater than 1.0)

If (ZeroCalibrationError is greater than or equal to 0 and less than or equal to 1.0)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferencePCT".

If (the absolute value of the difference between *diff* and ZeroCalibrationError is less than or equal to the Tolerance in the cross-check record)
if (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSED".

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than ZeroInjectionDate/ZeroInjectionHour)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)

If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than ZeroInjectionDate/ZeroInjectionHour)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.

else if (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSED".

If (ComponentTypeCode of the associated component is equal to "SO2" or "NOX")

Calculate ***Daily Cal Zero Injection Calc Result*** = min(round(*diff* / ***Daily Cal Span Value*** * 100, 1), 9999.9)
Round *diff* to 1 decimal places.

If (***Daily Cal Zero Injection Calc Result*** is greater than 5.0, AND (***Daily Cal Span Value*** is less than or equal to 50 AND *diff* is less than or equal to 5.0) OR (***Daily Cal Span Value*** is greater than 50 AND ***Daily Cal Span Value*** is less than or equal to 200 AND *diff* is less than or equal to 10.0)))

set ***Daily Cal Zero Injection Calc Result*** to *diff*.
set ***Daily Cal Zero Injection Calc APS Indicator*** to 1.

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "FAILED" or "INC" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSAPS".

Otherwise,

If (***Daily Cal Zero Injection Calc Result*** is greater than 5.0)

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "IGNORED")

If (ZeroAPSIndicator is NOT equal to 1 and ZeroCalibrationError is greater than or equal to 0 and less than or equal to 5.0)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "CalibrationError".

If (the absolute value of the difference between ***Daily Cal Zero Injection Calc Result*** and ZeroCalibrationError is less than or equal to the Tolerance in the cross-check record)
If (***Daily Cal Calc Result*** is not equal to "PASSAPS" or "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSED".

Otherwise,
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than ZeroInjectionDate/ZeroInjectionHour)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.

Otherwise,

If (ZeroAPSIndicator is equal to 1 and ZeroCalibrationError is greater than or equal to 0, and (***Daily Cal Span Value*** is less than or equal to 50 AND ZeroCalibrationError is less than or equal to 5.0) OR (***Daily Cal Span Value*** is greater than 50 AND ***Daily Cal Span Value*** is less than or equal to 200 AND ZeroCalibrationError is less than or equal to 10.0)))

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferencePPM".

If (the absolute value of the difference between *diff* and ZeroCalibrationError is less than or equal to the Tolerance in the cross-check record)
If (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSAPS".

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than ZeroInjectionDate/ZeroInjectionHour)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than ZeroInjectionDate/ZeroInjectionHour)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.

Otherwise,
If (***Daily Cal Calc Result*** is not equal to "FAILED" or "INC" or "PASSAPS" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSED".

If (ComponentTypeCode of the associated component is equal to "FLOW")

Calculate ***Daily Cal Zero Injection Calc Result*** = min(round(diff / ***Daily Cal Span Value*** * 100, 1), 9999.9). Round diff to 2 decimal places.

If (***Daily Cal Zero Injection Calc Result*** is greater than 6.0, the SampleAcquisitionMethodCode of the associated component is equal to "DP", and *diff* is less than or equal to 0.02)

set ***Daily Cal Zero Injection Calc Result*** to *diff*.
set ***Daily Cal Zero Injection Calc APS Indicator*** to 1.

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "FAILED" or "INC" or "IGNORED")

set ***Daily Cal Calc Result*** to "PASSAPS".

Otherwise,

If (***Daily Cal Zero Injection Calc Result*** is greater than 6.0)

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "IGNORED")

If (ZeroAPSIndicator is NOT equal to 1 and ZeroCalibrationError is greater than or equal to 0 and less than or equal to 6.0)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "CalibrationError".

If (the absolute value of the difference between ***Daily Cal Zero Injection Calc Result*** and ZeroCalibrationError is less than or equal to the Tolerance in the cross-check record)
If (***Daily Cal Calc Result*** is not equal to "PASSAPS" or "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSED".

Otherwise,
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than ZeroInjectionDate/ZeroInjectionHour)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.

Otherwise,

If (ZeroAPSIndicator is equal to 1, the SampleAcquisitionMethodCode of the associated component is equal to "DP", and ZeroCalibrationError is greater than or equal to 0 and less than or equal to 0.02)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferenceINH2O".

If (the absolute value of the difference between *diff* and ZeroCalibrationError is less than or equal to the Tolerance in the cross-check record)
If (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSAPS".

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is

greater than ZeroInjectionDate/ZeroInjectionHour)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than ZeroInjectionDate/ZeroInjectionHour)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.

Otherwise,
If (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED", or "PASSAPS" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSED".

If (ComponentTypeCode of the associated component is equal to "HG")

Calculate ***Daily Cal Zero Injection Calc Result*** = min(round(*diff* / ***Daily Cal Span Value*** * 100, 1), 9999.9)
Round *diff* to 1 decimal places.

If (***Daily Cal Zero Injection Calc Result*** is greater than 5.0, AND *diff* is less than or equal to 1.0)

set ***Daily Cal Zero Injection Calc Result*** to *diff*.
set ***Daily Cal Zero Injection Calc APS Indicator*** to 1.

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "FAILED" or "INC" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSAPS".

Otherwise,

If (***Daily Cal Zero Injection Calc Result*** is greater than 5.0)

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "IGNORED")

If (ZeroAPSIndicator is NOT equal to 1 and ZeroCalibrationError is greater than or equal to 0 and less than or equal to 5.0)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "CalibrationError".

If (the absolute value of the difference between ***Daily Cal Zero Injection Calc Result*** and ZeroCalibrationError is less than or equal to the Tolerance in the cross-check record)
If (***Daily Cal Calc Result*** is not equal to "PASSAPS" or "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSED".

Otherwise,
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)

set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than ZeroInjectionDate/ZeroInjectionHour)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.

Otherwise,

If (ZeroAPSIndicator is equal to 1 and ZeroCalibrationError is greater than or equal to 0 and less than or equal to 1.0)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "Difference UGSCM".

If (the absolute value of the difference between *diff* and ZeroCalibrationError is less than or equal to the Tolerance in the cross-check record)
If (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSAPS".

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than ZeroInjectionDate/ZeroInjectionHour)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than ZeroInjectionDate/ZeroInjectionHour)
set ***Daily Cal Fail Date*** to ZeroInjectionDate.
set ***Daily Cal Fail Hour*** to ZeroInjection Hour.

Otherwise,
If (***Daily Cal Calc Result*** is not equal to "FAILED" or "INC" or "PASSAPS" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSED".

Results:

Result	Response	Severity
A	The software could not evaluate the [test] calculations reported for [key], because of the errors listed above.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-21

Check Name: Calculate Upscale Gas Injection or Reference Signal Results

Related Former Checks:

Applicability: CEM Check

Description: This check is to calculate calibration errors.

Validation Tables:

Test Tolerances (Cross Check Table)

Specifications:

For the daily calibration test with an upscale injection:

If (***Daily Cal Span Value*** is null, or ***Daily Cal Upscale Gas Level Valid*** is false, or UpscaleReferenceValue of the test is null or is less than or equal to zero, or UpscaleMeasuredValue of the test is null)
Set ***Daily Cal Calc Result*** to "INVALID", ***Daily Cal Upscale Injection Calc Result*** to null, ***Daily Cal Upscale Injection Calc APS Indicator*** to null, and return result A.

Otherwise,

Calculate *diff* = abs(UpscaleMeasuredValue - UpscaleReferenceValue)
Set ***Daily Cal Upscale Injection Calc APS Indicator*** to 0.

If (ComponentTypeCode of the associated component is equal to "CO2" or "O2")

Round *diff* to 1 decimal place.
Set ***Daily Cal Upscale Injection Calc Result*** to *diff*.

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "IGNORED")

If (***Daily Cal Upscale Injection Calc Result*** is greater than 1.0)

If (UpscaleCalibrationError is greater than or equal to 0 and less than or equal to 1.0)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferencePCT".

If (the absolute value of the difference between *diff* and UpscaleCalibrationError is less than or equal to the Tolerance in the cross-check record)
if (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSED".

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than UpscaleInjectionDate/UpscaleInjectionHour)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than UpscaleInjectionDate/UpscaleInjectionHour)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

else if (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSED".

If (ComponentTypeCode of the associated component is equal to "SO2" or "NOX")

Calculate ***Daily Cal Upscale Injection Calc Result*** = min(round(*diff* / ***Daily Cal Span Value*** * 100, 1), 9999.9)
Round *diff* to 1 decimal places.

If (***Daily Cal Upscale Injection Calc Result*** is greater than 5.0, AND (***Daily Cal Span Value*** is less than or equal to 50 AND *diff* is less than or equal to 5.0) OR (***Daily Cal Span Value*** is greater than 50 AND ***Daily Cal Span Value*** is less than or equal to 200 AND *diff* is less than or equal to 10.0)))

set ***Daily Cal Upscale Injection Calc Result*** to *diff*.
set ***Daily Cal Upscale Injection Calc APS Indicator*** to 1.

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "FAILED" or "INC" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSAPS".

Otherwise,

If (***Daily Cal Upscale Injection Calc Result*** is greater than 5.0)

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "IGNORED")

If (UpscaleAPSIndicator is NOT equal to 1 and UpscaleCalibrationError is greater than or equal to 0 and less than or equal to 5.0)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "CalibrationError".

If (the absolute value of the difference between ***Daily Cal Upscale Injection Calc Result*** and UpscaleCalibrationError is less than or equal to the Tolerance in the cross-check record)
If (***Daily Cal Calc Result*** is not equal to "PASSAPS" or "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSED".

Otherwise,
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than UpscaleInjectionDate/UpscaleInjectionHour)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

Otherwise,

If (UpscaleAPSIndicator is equal to 1 and UpscaleCalibrationError is greater than or equal to 0, and (***Daily Cal Span Value*** is less than or equal to 50 AND UpscaleCalibrationError is less than or equal to 5.0) OR (***Daily Cal Span Value*** is greater than 50 AND ***Daily Cal Span Value*** is less than or equal to 200 AND UpscaleCalibrationError is less than or equal to 10.0)))

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferencePPM".

If (the absolute value of the difference between *diff* and UpscaleCalibrationError is less than or equal to the Tolerance in the cross-check record)
If (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSAPS".

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than UpscaleInjectionDate/UpscaleInjectionHour)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than UpscaleInjectionDate/UpscaleInjectionHour)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

Otherwise,
If (***Daily Cal Calc Result*** is not equal to "FAILED" or "INC", or "PASSAPS" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSED".

If (ComponentTypeCode of the associated component is equal to "FLOW")

Calculate ***Daily Cal Upscale Injection Calc Result*** = min(round(diff / ***Daily Cal Span Value*** * 100, 1), 9999.9). Round diff to 2 decimal places.

If (***Daily Cal Upscale Injection Calc Result*** is greater than 6.0, the SampleAcquisitionMethodCode of the associated component is equal to "DP", and *diff* is less than or equal to 0.02)

set ***Daily Cal Upscale Injection Calc Result*** to *diff*.
set ***Daily Cal Upscale Injection Calc APS Indicator*** to 1.

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "FAILED" or "INC" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSAPS".

Otherwise,

If (***Daily Cal Upscale Injection Calc Result*** is greater than 6.0)

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "IGNORED")

If (UpscaleAPSIndicator is NOT equal to 1 and UpscaleCalibrationError is greater than or equal to 0 and less than or equal to 6.0)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "CalibrationError".

If (the absolute value of the difference between ***Daily Cal Upscale Injection Calc Result*** and UpscaleCalibrationError is less than or equal to the Tolerance in the cross-check record)
If (***Daily Cal Calc Result*** is not equal to "PASSAPS" or "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSED".

Otherwise,
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than UpscaleInjectionDate/UpscaleInjectionHour)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

Otherwise,

If (UpscaleAPSIndicator is equal to 1, the SampleAcquisitionMethodCode of the associated component is equal to "DP", and UpscaleCalibrationError is greater than or equal to 0 and less than or equal to 0.02)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferenceINH2O".

If (the absolute value of the difference between *diff* and UpscaleCalibrationError is less than or equal to the Tolerance in the cross-check record)
If (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED")
set ***Daily Cal Calc Result*** to "PASSAPS".

else

set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than UpscaleInjectionDate/UpscaleInjectionHour)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

else
set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)
If (***Daily Cal Fail Date*** is null)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than UpscaleInjectionDate/UpscaleInjectionHour)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

Otherwise,
If (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED", or "PASSAPS" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSED".

If (ComponentTypeCode of the associated component is equal to "HG")

Calculate ***Daily Cal Upscale Injection Calc Result*** = min(round(*diff* / ***Daily Cal Span Value*** * 100, 1), 9999.9)
Round *diff* to 1 decimal places.

If (***Daily Cal Upscale Injection Calc Result*** is greater than 5.0, AND *diff* is less than or equal to 1.0)

set ***Daily Cal Upscale Injection Calc Result*** to *diff*.
set ***Daily Cal Upscale Injection Calc APS Indicator*** to 1.

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "FAILED" or "INC" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSAPS".

Otherwise,

If (***Daily Cal Upscale Injection Calc Result*** is greater than 5.0)

If (***Daily Cal Calc Result*** is not equal to "INVALID" or "IGNORED")

If (UpscaleAPSIndicator is NOT equal to 1 and UpscaleCalibrationError is greater than or equal to 0 and less than or equal to 5.0)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "CalibrationError".

If (the absolute value of the difference between ***Daily Cal Upscale Injection Calc Result*** and UpscaleCalibrationError is less than or equal to the Tolerance in the cross-check record)

If (***Daily Cal Calc Result*** is not equal to "PASSAPS" or "INC" or "FAILED")

set ***Daily Cal Calc Result*** to "PASSED".

Otherwise,

set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)

If (***Daily Cal Fail Date*** is null)

set ***Daily Cal Fail Date*** to UpscaleInjectionDate.

set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than UpscaleInjectionDate/UpscaleInjectionHour)

set ***Daily Cal Fail Date*** to UpscaleInjectionDate.

set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

Otherwise,

If (UpscaleAPSIndicator is equal to 1 and UpscaleCalibrationError is greater than or equal to 0 AND UpscaleCalibrationError is less than or equal to 1.0)

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferenceUGSCM".

If (the absolute value of the difference between *diff* and UpscaleCalibrationError is less than or equal to the Tolerance in the cross-check record)

If (***Daily Cal Calc Result*** is not equal to "INC" or "FAILED")

set ***Daily Cal Calc Result*** to "PASSAPS".

else

set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)

If (***Daily Cal Fail Date*** is null)

set ***Daily Cal Fail Date*** to UpscaleInjectionDate.

set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than UpscaleInjectionDate/UpscaleInjectionHour)

set ***Daily Cal Fail Date*** to UpscaleInjectionDate.

set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

else

set ***Daily Cal Calc Result*** to "FAILED".

If (***Daily Cal Injection Times Valid*** == true)

If (***Daily Cal Fail Date*** is null)

set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.
else if (***Daily Cal Fail Date/Daily Cal Fail Hour*** is greater than UpscaleInjectionDate/UpscaleInjectionHour)
set ***Daily Cal Fail Date*** to UpscaleInjectionDate.
set ***Daily Cal Fail Hour*** to UpscaleInjection Hour.

Otherwise,
If (***Daily Cal Calc Result*** is not equal to "FAILED" or "INC", or "PASSAPS" or "IGNORED")
set ***Daily Cal Calc Result*** to "PASSED".

Results:

Result	Response	Severity
A	The software could not evaluate the [test] calculations reported for [key], because of the errors listed above.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-22

Check Name: Daily Calibration Test End Time Valid

Related Former Checks:

Applicability: CEM Check

Description: This check indicates if the reported test end date, hour, minute is consistent with the injection times.

Specifications:

For the daily calibration test with upscale and zero injections and a valid date, hour, and minute and injection times:

If Date, Hour, and Minute of the test does not equal the later of the ZeroInjectionDate, Hour, and Minute (if not null) and the UpscaleInjectionDate, Hour, and Minute (if non-null),
return result A.

Results:

Result	Response	Severity
A	You reported a test Date, Hour, and Minute that is not the same as the Date, Hour, and Minute of the last injection in the daily calibration test for [key].	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-23

Check Name: Reported Zero Injection Results Consistent with Recalculated Values

Related Former Checks:

Applicability: CEM Check

Description: This check is to compare reported and recalculated results for each gas injection.

Validation Tables:

Test Tolerances (Cross Check Table)

Specifications:

For the daily calibration test with a zero injection:

If the ZeroAPSIndicator is equal to 1, the ComponentTypeCode of the associated component is equal to "FLOW", and the SampleAcquisitionMethodCode of the associated component is not equal to "DP",
return result A.

If the ZeroAPSIndicator is equal to 1, the ComponentTypeCode of the associated component is equal to "SO2" or "NOX", and the ***Daily Cal Span Value*** is greater than or equal to 200,
return result B.

If the ZeroAPSIndicator is equal to 1, the ComponentTypeCode of the associated component is equal to "CO2" or "O2",
return result C.

Otherwise,

If ***Daily Cal Zero Injection Calc Result*** is not null,

If the ZeroAPSIndicator in the current record is not equal to 1 and the ***Daily Cal Zero Injection Calc APS Indicator*** is equal to 1,
return result D.

If the ZeroCalibrationError is greater than or equal to 0,

If the ComponentTypeCode of the associated component is equal to "CO2" or "O2"

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferencePCT".

If the absolute value of the difference between the ***Daily Cal Zero Injection Calc Result*** and the ZeroCalibrationError is greater than the Tolerance in the cross-check record,
return result E.

If the ***Daily Cal Zero Injection Calc APS Indicator*** is equal to 1,

If the ComponentTypeCode of the associated component is equal to "FLOW",

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferenceINH2O".

If the absolute value of the difference between the ***Daily Cal Zero Injection Calc Result*** and the ZeroCalibrationError is greater than the Tolerance in the cross-check record,
return result E.

Else, If the ComponentTypeCode of the associated component is equal to "HG",

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to

"7DAY" and the FieldDescription is equal to "DifferenceUGSCM".

If the absolute value of the difference between the ***Daily Cal Zero Injection Calc Result*** and the ZeroCalibrationError is greater than the Tolerance in the cross-check record,
return result E.

Otherwise,

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferencePPM".

If the absolute value of the difference between the ***Daily Cal Zero Injection Calc Result*** and the ZeroCalibrationError is greater than the Tolerance in the cross-check record,
return result E.

else if ZeroAPSIndicator is equal to 0,

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "CalibrationError".

If the absolute value of the difference between the ***Daily Cal Zero Injection Calc Result*** and the ZeroCalibrationError is greater than the Tolerance in the cross-check record,
return result F.

Results:

Result	Response	Severity
A	You reported a value of "1" as the [level] APS Indicator for [key], but you must use the standard performance criteria for non-differential pressure flow monitors.	Critical Error Level 1
B	You reported a value of "1" as the [level] APS Indicator for [key], but you must use the standard performance specification criteria for SO2 and NOX components when the instrument span is greater than or equal to 200.	Critical Error Level 1
C	You reported a value of "1" as the [level] APS Indicator for [key], but you must use the standard performance specification criteria for CO2 and O2 components.	Critical Error Level 1
D	You did not report a value of "1" in the [level] APS Indicator for [key], although EPA applied the alternative performance specification to determine that the injection passed the applicable performance specification.	Critical Error Level 1
E	The absolute difference reported as the [level] Calibration Error for [key] is inconsistent with the recalculated absolute difference for the gas injection or reference signal.	Critical Error Level 1
F	The [level] Calibration Error reported for [key] is inconsistent with the recalculated calibration error for the gas injection or reference signal.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-24

Check Name: Reported Upscale Injection Results Consistent with Recalculated Values

Related Former Checks:

Applicability: CEM Check

Description: This check is to compare reported and recalculated results for each gas injection.

Validation Tables:

Test Tolerances (Cross Check Table)

Specifications:

For the daily calibration test with an upscale injection:

If the UpscaleAPSIndicator is equal to 1, the ComponentTypeCode of the associated component is equal to "FLOW", and the SampleAcquisitionMethodCode of the associated component is not equal to "DP",
return result A.

If the UpscaleAPSIndicator is equal to 1, the ComponentTypeCode of the associated component is equal to "SO2" or "NOX", and the ***Daily Cal Span Value*** is greater than or equal to 200,
return result B.

If the UpscaleAPSIndicator is equal to 1, the ComponentTypeCode of the associated component is equal to "CO2" or "O2",
return result C.

Otherwise,

If ***Daily Cal Upscale Injection Calc Result*** is not null,

If the UpscaleAPSIndicator in the current record is not equal to 1 and the ***Daily Cal Upscale Injection Calc APS Indicator*** is equal to 1,
return result D.

If the UpscaleCalibrationError is greater than or equal to 0,

If the ComponentTypeCode of the associated component is equal to "CO2" or "O2"

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferencePCT".

If the absolute value of the difference between the ***Daily Cal Upscale Injection Calc Result*** and the UpscaleCalibrationError is greater than the Tolerance in the cross-check record,
return result E.

If the ***Daily Cal Upscale Injection Calc APS Indicator*** is equal to 1,

If the ComponentTypeCode of the associated component is equal to "FLOW",

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferenceINH2O".

If the absolute value of the difference between the ***Daily Cal Upscale Injection Calc Result*** and the UpscaleCalibrationError is greater than the Tolerance in the cross-check record,
return result E.

Else, if the ComponentTypeCode of the associated component is equal to "HG",

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferenceUGSCM".

If the absolute value of the difference between the ***Daily Cal Upscale Injection Calc Result*** and the UpscaleCalibrationError is greater than the Tolerance in the cross-check record,
return result E.

Otherwise,

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "DifferencePPM".

If the absolute value of the difference between the ***Daily Cal Upscale Injection Calc Result*** and the UpscaleCalibrationError is greater than the Tolerance in the cross-check record,
return result E.

else if UpscaleAPSIndicator is equal to 0,

Locate the Test Tolerance cross-check record where the TestTypeCode is equal to "7DAY" and the FieldDescription is equal to "CalibrationError".

If the absolute value of the difference between the ***Daily Cal Upscale Injection Calc Result*** and the UpscaleCalibrationError is greater than the Tolerance in the cross-check record,
return result F.

Results:

Result	Response	Severity
A	You reported a value of "1" as the [level] APS Indicator for [key], but you must use the standard performance criteria for non-differential pressure flow monitors.	Critical Error Level 1
B	You reported a value of "1" as the [level] APS Indicator for [key], but you must use the standard performance specification criteria for SO2 and NOX components when the instrument span is greater than or equal to 200.	Critical Error Level 1
C	You reported a value of "1" as the [level] APS Indicator for [key], but you must use the standard performance specification criteria for CO2 and O2 components.	Critical Error Level 1
D	You did not report a value of "1" in the [level] APS Indicator for [key], although EPA applied the alternative performance specification to determine that the injection passed the applicable performance specification.	Critical Error Level 1
E	The absolute difference reported as the [level] Calibration Error for [key] is inconsistent with the recalculated absolute difference for the gas injection or reference signal.	Critical Error Level 1
F	The [level] Calibration Error reported for [key] is inconsistent with the recalculated calibration error for the gas injection or reference signal.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-25

Check Name: Determination of Overall Daily Calibration Test Result

Related Former Checks:

Applicability: CEM Check

Description: This check is to calculate daily calibration test results.

Specifications:

For the daily calibration test:

If ***Daily Cal Calc Result*** is equal to "INVALID",
set ***Daily Cal Calc Result*** to null.

If TestResultCode is null,
return result A.

If TestResultCode is not equal to "PASSED", "PASSAPS", "FAILED", "INC", or "ABORTED",
return result B.

If ***Daily Cal Calc Result*** is equal to "FAILED",

If TestResultCode is equal to "PASSED" or "PASSAPS",
return result C.

If TestResultCode is equal to "INC",
return result D.

If ***Daily Cal Calc Result*** is equal to "PASSED" or "PASSAPS", and the TestResultCode is equal to "FAILED",
return result E.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	You reported the value [value], which is not in the list of valid values for this test type, in the field [fieldname] for [key].	Critical Error Level 1
C	The TestResultCode for [key] indicates a passing test, but the recalculated results indicate a failing test.	Critical Error Level 1
D	The TestResultCode for [key] indicates an incomplete test, but the recalculated results indicate a failing test. A test is considered to have failed if it fails to meet the performance criteria for any injection.	Critical Error Level 1
E	You reported a TestResultCode of "FAILED" for [key], but the results recalculated or determined from the other reported values indicate that the test passed.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-26

Check Name: Upscale Gas Type Code Valid

Related Former Checks:

Applicability: CEM Check

Description: This check determines whether the Upscale Gas Type Code is valid (PGVP).

Validation Tables:

Gas Type Code (Lookup Table)
Vw System Parameter (Lookup Table)

Specifications:

For the daily calibration test with an upscale injection:

UpscaleGasTypeValid = true.

Locate ***System Parameter*** lookup table record where Sys_Param_Name = 'PGVP_AETB_RULE_DATE'.

Set ***Daily Cal PGVP Rule Date*** to ***System Parameter***.Param_Value1.

If UpscaleGasTypeCode is null,

If the ComponentTypeCode of the associated component is equal to "SO2", "NOX", "CO2", or "O2", and the Date of the test is on or after 9/26/2011,
UpscaleGasTypeValid = false.
return result A.

Otherwise,

If the ComponentTypeCode of the associated component is equal to "FLOW", "HCL", or "HG",
UpscaleGasTypeValid = false
return result B.

else if the UpscaleGasTypeCode is not equal to "GMIS", "PRM", "RGM", or "SRM",

if the UpscaleGasTypeCode is not in the ***GasTypeCode lookup table***.
UpscaleGasTypeValid = false
return result C.

else if the UpscaleGasTypeCode == "ZERO" or "ZAM"
UpscaleGasTypeValid = false
return result C

else if the UpscaleGasTypeCode == "APPVD"
return result D

else if the ComponentTypeCode == "SO2", "NOX", "CO2" or "O2",

Locate ***Protocol Gas Parameter To Type Cross Reference*** records where ProtocolGasParameter is equal to ComponentTypeCode in the current Daily Calibration record, and GasTypeList contains the UpscaleGasTypeCode in the current Daily Calibration record.

If not found,
UpscaleGasTypeValid = false
return result E.

else if ComponentTypeCode == "O2", UpscaleGasTypeCode == "AIR", and the UpscaleGasCode is not

equal to "HIGH",
UpscaleGasTypeValid = false
return result F.

Results:

Result	Response	Severity
A	You did not report a UpscaleGasTypeCode for [key]. This information is required by the Protocol Gas Verification Program reporting rule.	Critical Error Level 1
B	You reported a value in the UpscaleGasTypeCode field for [key]. This value should not be reported for a FLOW, HCl, or HG component.	Critical Error Level 1
C	You reported the value [value], which is not in the list of valid values, in the field [fieldname] for [key].	Critical Error Level 1
D	You reported "APPVD" as the [fieldname] for [key]. This code indicates that you received approval from EPA for a new type of Protocol Gas. If you have not received approval from EPA, please contact ECMPS support. If you have already received approval, you should log in to the ECMPS host, so that the ECMPS program can obtain the necessary information to override this error.	Critical Error Level 1
E	You reported an UpscaleGasTypeCode that is not appropriate for a [comptype] component for [key].	Critical Error Level 1
F	You reported an [fieldname] of "AIR" for [key], which indicates the use of purified air material, but this material can only be used for a high-level calibration.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-27

Check Name: Cylinder ID Valid

Related Former Checks:

Applicability: CEM Check

Description: This check determines whether the Cylinder ID is valid.

Specifications:

For the daily calibration test with an upscale injection:

If CylinderID is null,
If ***UpscaleGasTypeValid*** is true, and the UpscaleGasTypeCode is not null and not equal to "AIR",
return result A.

Otherwise,
If the UpscaleGasTypeCode is equal to "AIR",
return result B.

else if ***UpscaleGasTypeValid*** is true, and the UpscaleGasTypeCd is null,
return result C.

else if CylinderIdentifer contains characters that are not capital letters, numbers, hyphens, ampersands or periods,
If ***InvalidCylinderIdList*** does not contain CylinderID,
Add CylinderID to ***InvalidCylinderIdList***.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	You indicated that you used purified air material or zero air material instead of a cylinder gas, but you reported a CylinderIdentifier.	Critical Error Level 1
C	You reported a [fieldname] for [key], but you did not report an UpscaleGasTypeCode.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-28

Check Name: Vendor ID Valid

Related Former Checks:

Applicability: CEM Check

Description: This check determines whether the Vendor ID is valid.

Validation Tables:

Protocol Gas Vendor (Lookup Table)

Specifications:

For the daily calibration test with an upscale injection:

If VendorID is null,

If ***UpscaleGasTypeValid*** is true, and the UpscaleGasTypeCode is not null and not equal to "AIR", "SRM", "NTRM", "GMIS", "RGM", or "PRM",

return result A.

else if the VendorID is not in the ***Protocol Gas Vendor*** lookup table,

return result B.

Otherwise,

If the UpscaleGasTypeCode is equal to "AIR", "SRM", "NTRM", "GMIS", "RGM", or "PRM",

return result C.

else if the DeactivationDate in the ***Protocol Gas Vendor*** record is not null and the Date of the current test is on or after the January 1 after DeactivationDate + 8 years,

return result F.

else if the ActivationDate in the **Protocol Gas Vendor** record is after the Date of the current test,

return result G.

else if the VendorID is equal to "NONPGVP", and the Date of the test is on or after the ***Daily Cal PGVP Rule Date*** + 60 days + 8 years,

return result D.

else if ***UpscaleGasTypeValid*** is true, and the UpscaleGasTypeCd is null,

return result E.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	You reported a VendorIdentifier of [value], which is not in the list of Protocol Gas Vendors, for [key]. Please visit the ECMPS Support Website for the list of Protocol Gas Vendors.	Critical Error Level 1
C	You reported a [fieldname] for [key], but this value should only be reported for an EPA Protocol Gas Type. The cylinder gas type of [gastype] indicates the use of a non-EPA Protocol Gas Type.	Critical Error Level 1
D	You reported a VendorIdentifier of "NONPGVP" for [key], indicating the use of a EPA Protocol Gas Type purchased from a vendor not participating in the Protocol Gas Vendor Program (PGVP). You cannot use a gas purchased from a non-participating vendor that was acquired more than 60 days after the PGVP Effective Date.	Critical Error Level 2
E	You reported a [fieldname] for [key], but you did not report an UpscaleGasTypeCode.	Non-Critical Error
F	You have reported a VendorIdentifier for [key] of a vendor who is no longer participating in the Protocol Gas Verification Program.	Critical Error Level 2
G	You have reported a vendor for [key], but this vendor was not active at the time of this test.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-29

Check Name: Cylinder Expiration Date Valid

Related Former Checks:

Applicability: CEM Check

Description: This check determines whether the Expiration Date of the cylinder is valid.

Specifications:

For the daily calibration test with an upscale injection:

If ExpirationDate is null,
If ***UpscaleGasTypeValid*** is true, and the UpscaleGasTypeCode is not null and not equal to "AIR",
return result A.

Otherwise,
If the UpscaleGasTypeCode is equal to "AIR",
return result B.

else if the ExpirationDate is prior to the Date of the test,
return result C.

else if the ExpirationDate is more than 8 years after the Date of the test,
return result D.

else if ***UpscaleGasTypeValid*** is true, and the UpscaleGasTypeCd is null,
return result E.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	You reported a [fieldname] for [key], but this value should only be reported for an EPA Protocol Gas Type. The cylinder gas type of [gastype] indicates the use of a non-EPA Protocol Gas Type.	Critical Error Level 1
C	You reported an ExpirationDate for the cylinder that is prior to the date of the test for [key].	Critical Error Level 2
D	You reported an ExpirationDate for the cylinder that is more than eight years after the date of the test for [key]. Gas cylinders expire in less than eight years.	Critical Error Level 2
E	You reported a [fieldname] for [key], but you did not report an UpscaleGasTypeCode.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Code: DAYCAL-30

Check Name: Upscale Gas Type Code Components Valid

Related Former Checks: 2013 Q1 replacement for DAYCAL-26

Applicability: CEM Check

Description: This check determines whether the Gas Components in the Upscale Gas Type Code are valid (PGVP).

Validation Tables:

Vw System Parameter (Lookup Table)

Specifications:

For the daily calibration test with an upscale injection:

UpscaleGasTypeValid = true.

Locate ***System Parameter*** lookup table record where Sys_Param_Name = 'PGVP_AETB_RULE_DATE'.

Set ***Daily Cal PGVP Rule Date*** to ***System Parameter***.Param_Value1.

If UpscaleGasTypeCode is null,

If the ComponentTypeCode of the associated component is equal to "SO2", "NOX", "CO2", or "O2", and the Date of the test is on or after 9/26/2011,
UpscaleGasTypeValid = false.
return result A.

Otherwise,

If the ComponentTypeCode of the associated component is equal to "FLOW", "HCL", or "HG",
UpscaleGasTypeValid = false
return result B.

Else

Set ***Protocol Gas Invalid Component List*** to null.
Set ***Protocol Gas Exclusive Component List*** to null.
Set ***Protocol Gas Balance Component List*** to null.
Set ***Protocol Gas Duplicate Component List*** to null.

Set *Protocol Gas Component List* to null.
Set *Protocol Gas Approval Requested* = true.
Set *Protocol Gas Component Count* to 0.
Set *Balance Component Count* to 0.

For each GasComponentCode in UpscaleGasTypeCode ,

Locate a record in the ***GasComponentCodeLookupTable*** where GasComponentCode is equal to the GasComponentCode in the UpscaleGasTypeCode .

If not found, or GasComponentCode is equal to "ZERO",
Add GasComponentCode to ***Protocol Gas Invalid Component List***.

Else
If CanCombineIndicator is equal to 0,
Add GasComponentCode to ***Protocol Gas Exclusive Component List***.

If BalanceComponentIndicator is equal to 1,
Add GasComponentCode to ***Protocol Gas Balance Component List***.
Increament *Balance Component Count* by 1.

If the GasComponentCode is equal to "APPVD",
Set *Protocol Gas Approval Requested* = true.

If GasComponentCode is not in *Protocol Gas Component List*,
add GasComponentCode to *Protocol Gas Component List.*
Else if GasComponentCode is not in ***Protocol Gas Duplicate Component List***,
add GasComponentCode to ***Protocol Gas Duplicate Component List***.

Increament *Protocol Gas Component Count* by 1.

If ***Protocol Gas Invalid Component List*** is not null,
UpscaleGasTypeValid = false
return result C.

Else if ***Protocol Gas Duplicate Component List*** is not null,
set ***Protocol Gas Component List Valid*** = false.
return result L.

Else if ***Protocol Gas Exclusive Component List*** is not null, and *Protocol Gas Component Count* is greater than 1,
UpscaleGasTypeValid = false
return result D.

Else if ***Protocol Gas Approval Requested*** is equal to true,
return result E.

Else if ***Protocol Gas Exclusive Component List*** is null, and *Balance Component Count* is equal to 0,

set ***UpscaleGasTypeValid*** = false.
return result J.

Else if ***Protocol Gas Exclusive Component List*** is null, and *Balance Component Count* is greater than 1,

set ***UpscaleGasTypeValid*** = false.
return result K.

Else if the UpscaleGasTypeCode is not equal to "GMIS", "NTRM", "PRM", "RGM", or "SRM",

If the ComponentTypeCode is equal to "SO2" or "CO2",

If no GasComponentCode in UpscaleGasTypeCode is equal to ComponentTypeCode,
UpscaleGasTypeValid = false
return result F.

Else if the ComponentTypeCode is equal to "O2",

If UpscaleGasTypeCode is not equal to "AIR", and no GasComponentCode in UpscaleGasTypeCode is equal to "O2",
UpscaleGasTypeValid = false
return result G.

Else if UpscaleGasTypeCode is equal to "AIR", and the UpscaleGasCode is not equal to "HIGH",
UpscaleGasTypeValid = false

return result H.

Else if the ComponentTypeCode is equal to "NOX",

If no GasComponentCode in GasTypeCode is equal to "NO", "NO2", or "NOX",
UpscaleGasTypeValid = false
return result I.

Results:

Result	Response	Severity
A	You did not report a UpscaleGasTypeCode for [key]. This information is required by the Protocol Gas Verification Program reporting rule.	Critical Error Level 1
B	You reported a value in the UpscaleGasTypeCode field for [key]. This value should not be reported for a FLOW, HCl, or HG component.	Critical Error Level 1
C	You reported the values ([invalidlist]), in the field [fieldname] for [key], which are not in the list of valid values.	Critical Error Level 1
D	You reported multiple gas components in the field [fieldname] for [key] that include values ([exclusivelist]) that you should report by themselves.	Critical Error Level 1
E	You reported "APPVD" as the [fieldname] for [key]. This code indicates that you received approval from EPA for a new type of Protocol Gas. If you have not received approval from EPA, please contact ECMPS support. If you have already received approval, you should log in to the ECMPS host, so that the ECMPS program can obtain the necessary information to override this error.	Critical Error Level 1
F	You reported an UpscaleGasTypeCode that is not appropriate for a [comptype] component for [key].	Critical Error Level 1
G	You reported an UpscaleGasTypeCode that is not appropriate for a [comptype] component for [key].	Critical Error Level 1
H	You reported an [fieldname] of "AIR" for [key], which indicates the use of purified air material, but this material can only be used for a high-level calibration.	Critical Error Level 1
I	You reported an UpscaleGasTypeCode that is not appropriate for a [comptype] component for [key].	Critical Error Level 1
J	You reported an UpscaleGasTypeCode that does not contain a PGVP balance component. A single balance component is required when reporting other individual gas components.	Critical Error Level 1
K	You reported an UpscaleGasTypeCode that contains multiple PGVP balance components ([balancelist]). A single balance component is required when reporting other individual gas components.	Critical Error Level 1
L	Your reported one or more duplicate gas component records.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Daily Calibration Test

Check Category:

Daily Emissions Data

Check Code: DAILY-1

Check Name: Determine Need for Daily CO2 Emissions Record

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Current CO2 Mass Daily Record = null

Daily Op Time = null
if (***Daily Op Time Accumulator Array*** for the location >= 0)
    Daily Op Time = ***Daily Op Time Accumulator Array*** for the location

Daily Op Time Accumulator Array for the location = 0.

CO2 Method Count = Active records in MonitoringMethodData for the location and date where
    ParameterCode begins with "CO2"

FSA Method Count = Active records in MonitoringMethodData for the location and date where
    ParameterCode = "CO2M" and MethodCode = "FSA"

if (*FSA Method Count* > 0 AND *CO2 Method Count* > 1)
    return result A
else
    if (*FSA Method Count* > 0)
        Expected Summary Value for CO2 Array for the location = true

CO2 Mass Daily Emissions Count = count of DailyEmissionsData records with ParameterCode = "CO2M" where
        Current Date = DailyEmissionsData.Date

    if (*CO2 Mass Daily Emissions Count* > 1)
        Rpt Period CO2 Mass Reported Accumulator Array for the location = -1
        Rpt Period CO2 Mass Calculated Accumulator Array for the location = -1
        return result B

    else if (*FSA Method Count* == 0 AND *CO2 Mass Daily Emissions Count* > 0)
        return result C

    else if (*FSA Method Count* > 0 AND *CO2 Mass Daily Emissions Count* == 0 AND *Daily Op Time* > 0
        return result D

    else if (*FSA Method Count* > 0 AND *CO2 Mass Daily Emissions Count* == 1)
        Current CO2 Mass Daily Record = matching DailyEmissionsData record

        If (***Daily Op Time*** == 0)
            return result E

Results:

Result	Response	Severity
A	You have reported more than one active method to determine CO2 emissions in your monitoring plan for this date.	Critical Error Level 1
B	You reported more than one Daily Emissions record for [param] for the day.	Critical Error Level 1
C	You reported a Daily Emissions record for CO2M, but you did not report an active CO2M FSA method record in your monitoring plan for the day.	Critical Error Level 1
D	You did not report a Daily Emissions record for CO2M for the day.	Critical Error Level 1
E	You reported a Daily Emissions record for CO2M, but this is not appropriate for a non-operating day.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Daily Emissions

Check Code: DAILY-2

Check Name: Check Total Daily Emissions Value

Related Former Checks:

Applicability:

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

Set ***Calc TDE*** to null.

If (***Current CO2 Mass Daily Record*** is not null)

If (***Current CO2 Mass Daily Record***.TotalDailyEmissions >= 0 AND ***Rpt Period CO2 Mass Reported Accumulator Array*** for the location >= 0)
Rpt Period CO2 Mass Reported Accumulator Array for the location = ***Rpt Period CO2 Mass Reported Accumulator Array*** for the location + ***Current CO2 Mass Daily Record***.TotalDailyEmissions

if (***Current CO2 Mass Daily Record***.UnadjustedDailyEmissions is not null OR ***Calc CO2 Unadj*** is not null)

if (***Current CO2 Mass Daily Record***.AdjustedDailyEmissions is null)
Calc TDE = ***Calc CO2 Unadj***

else if (***Current CO2 Mass Daily Record***.UnadjustedDailyEmissions is not null AND ***Current CO2 Mass Daily Record***.AdjustedDailyEmissions is greater than or equal to 0 AND is less than or equal to ***Current CO2 Mass Daily Record***.UnadjustedDailyEmissions)
Calc TDE = ***Current CO2 Mass Daily Record***.AdjustedDailyEmissions

else if (***Current CO2 Mass Daily Record***.AdjustedDailyEmissions is greater than or equal to 0)
Calc TDE = ***Current CO2 Mass Daily Record***.AdjustedDailyEmissions

If (***Calc TDE*** is not null)

If (***Current CO2 Mass Daily Record***.SorbentRelatedMassEmissions is not null)

If (***Current CO2 Mass Daily Record***.SorbentRelatedMassEmissions >= 0)
Calc TDE = ***Calc TDE*** + ***Current CO2 Mass Daily Record***.SorbentRelatedMassEmissions

else
Set ***Calc TDE*** to null.

else if (***Current CO2 Mass Daily Record***.UnadjustedDailyEmissions is null AND ***Legacy Data Evaluation*** == true AND ***Current CO2 Mass Daily Record***.TotalDailyEmissions >= 0)

Calc TDE = ***Current CO2 Mass Daily Record***.TotalDailyEmissions

If (***Calc TDE*** is null)
Rpt Period CO2 Mass Calculated Accumulator Array for the location = -1

else
If (***Rpt Period CO2 Mass Calculated Accumulator Array*** for the location >= 0)
Rpt Period CO2 Mass Calculated Accumulator Array for the location = ***Rpt Period CO2 Mass Calculated Accumulator Array*** for the location + ***Calc TDE***

If (***Current CO2 Mass Daily Record***.TotalDailyEmissions >= 0)

If (***Calc TDE*** is not null)

Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "CO2M DAILY" AND
UOM = "TON"

if (ABS(***Current CO2 Mass Daily Record***.TotalDailyEmissions - ***Calc TDE***) > *Tolerance*)
return result A

else
return result C

else
Rpt Period CO2 Mass Reported Accumulator Array for the location = -1
return result B

Results:

Result	Response	Severity
A	The [fieldname] reported in the Daily Emissions record for [param] is inconsistent with the recalculated value.	Critical Error Level 1
B	The [fieldname] reported in the Daily Emissions record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
C	The TotalDailyEmissions in the Daily Emissions record for [param] could not be recalculated due to other errors listed in this report.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2 Daily Emissions

Check Code: DAILY-3

Check Name: Check Adjusted Daily Emissions Value

Related Former Checks:

Applicability:

Description:

Specifications:

if (***Current CO2 Mass Daily Record*** is not null),

if (***Current CO2 Mass Daily Record***.AdjustedDailyEmissions is not null),

If (***Current CO2 Mass Daily Record***.AdjustedDailyEmissions is less than 0),
return result A

else
If (***Current CO2 Mass Daily Record***.UnadjustedDailyEmissions is greater than or equal to 0 AND is less than ***Current CO2 Mass Daily Record***.AdjustedDailyEmissions),
return result B

Results:

Result	Response	Severity
A	The [fieldname] reported in the Daily Emissions record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
B	The AdjustedDailyEmissions in the Daily Emissions record for [param] is greater than the UnadjustedDailyEmissions. The adjusted value should be less than the unadjusted value.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2 Daily Emissions

Check Code: DAILY-4

Check Name: Check Sorbent Related Emissions

Related Former Checks:

Applicability:

Description:

Specifications:

if (***Current CO2 Mass Daily Record*** is not null),

if (***Current CO2 Mass Daily Record***.SorbentRelatedMassEmissions is not null AND is less than 0),
return result A

Results:

Result	Response	Severity
A	The [fieldname] reported in the Daily Emissions record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2 Daily Emissions

Check Code: DAILY-5

Check Name: Validate Presence of Adjusted Daily Emissions

Related Former Checks:

Applicability:

Description:

Specifications:

if (***Current CO2 Mass Daily Record***) is not null

if (***Current CO2 Mass Daily Record***.AdjustedDailyEmissions is not null)

Locate a Monitor Formula record for the location and hour where the ParameterCode is equal to 'CO2M" and the FormulaCode is equal to "G-2" or "G-3".

If not found,
return result A.

Results:

Result	Response	Severity
A	You reported AdjustedDailyEmissions in the Daily Emissions record for CO2M, but you did not report a G-2 or G-3 formula in your monitoring plan.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2 Daily Emissions

Check Code: DAILY-6

Check Name: Validate Presence of Sorbent Related Emissions

Related Former Checks:

Applicability:

Description:

Specifications:

if (***Current CO2 Mass Daily Record***) is not null

if (***Current CO2 Mass Daily Record***.SorbentRelatedMassEmissions is not null),

Missing CO2M Formula = null

Locate a *Monitor Formula* record for the location and hour where the ParameterCode is equal to 'CO2M" and the FormulaCode is equal to "G-5" or "G-6".

If not found,
Set ***Missing CO2M Formula*** to "G-5 or G-6"

Locate a *Monitor Formula* record for the location and hour where the ParameterCode is equal to 'CO2M" and the FormulaCode is equal to "G-8".

If not found,
Append "G-8" to ***Missing CO2M Formula***.

If (***Missing CO2M Formula*** is not null)
return result A.

Results:

Result	Response	Severity
A	You reported SorbentRelatedMassEmissions in the Daily Emissions record for CO2M, but you did not report [code] formula(s) in your monitoring plan.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2 Daily Emissions

Check Code: DAILY-7

Check Name: Check Unadjusted Daily Emissions Value

Related Former Checks:

Applicability:

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

Set ***Calc CO2 Unadj*** to null.

if (***Current CO2 Mass Daily Record*** is not null)

if (***Calc Total Carbon Burned*** is greater than 0)

Calculate ***Calc CO2 Unadj*** = ***Calc Total Carbon Burned*** * 44 / 24,000, and round the result to 1 decimal place.

if (***Current CO2 Mass Daily Record***.UnadjustedDailyEmissions is null OR is less than 0)
return result A

else if (***Current CO2 Mass Daily Record***.UnadjustedDailyEmissions >= 0)

Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "CO2M DAILY" AND
UOM = "TON"

if (ABS(***Current CO2 Mass Daily Record***.UnadjustedDailyEmissions - ***Calc CO2 Unadj***) > *Tolerance*)
return result B

else if (***Current CO2 Mass Daily Record***.UnadjustedDailyEmissions is null)
If (***Legacy Data Evaluation*** == false)
return result A.

else if (***Current CO2 Mass Daily Record***.UnadjustedDailyEmissions is less than 0)
return result A

else
Set ***Calc CO2 Unadj*** to ***Current CO2 Mass Daily Record***.UnadjustedDailyEmissions.

Results:

Result	Response	Severity
A	The [fieldname] reported in the Daily Emissions record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
B	The [fieldname] reported in the Daily Emissions record for [param] is inconsistent with the recalculated value.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2 Daily Emissions

Check Code: DAILY-8

Check Name: Check Fuel in Daily Fuel Record

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Locate UnitFuel record for the location and day
where FuelCd = ***Current Daily Fuel Record***.UnitFuelCd

If not found,
return result A

Results:

Result	Response	Severity
A	You did not report an active Unit Fuel record for FuelCode [fuelcd] in your monitoring plan.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Daily Emissions Fuel

Check Code: DAILY-9

Check Name: Check Daily Fuel Feed

Related Former Checks:

Applicability:

Description:

Specifications:

if (***Current Daily Fuel Record***.DailyFuelFeed is null)
    return result A.

else if (***Current Daily Fuel Record***.DailyFuelFeed is less than or equal to 0)
    return result A.

Results:

Result	Response	Severity
A	The [fieldname] reported in the Daily Fuel record for [key] is missing or invalid. The value must be greater than 0.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Daily Emissions Fuel

Check Code: DAILY-10

Check Name: Check Carbon Content Used

Related Former Checks:

Applicability:

Description:

Specifications:

if (***Current Daily Fuel Record***.CarbonContentUsed is null)
return result A.

else if (***Current Daily Fuel Record***.CarbonContentUsed is less than or equal to 0 or greater than 100)
return result A.

Results:

Result	Response	Severity
A	The CarbonContentUsed in the Daily Fuel record for [key] is invalid. The value must be greater than 0 and less than or equal to 100.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Daily Emissions Fuel

Check Code: DAILY-11

Check Name: Check Fuel Carbon Burned

Related Former Checks:

Applicability:

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

Calc Fuel Carbon Burned = null

if (***Current Daily Fuel Record***.DailyFuelFeed is greater than 0 and ***Current Daily Fuel Record***.CarbonContentUsed is greater than 0 and less than or equal to 100)

Calculate ***Calc Fuel Carbon Burned*** = ***Current Daily Fuel Record***.DailyFuelFeed * ***Current Daily Fuel Record***.CarbonContentUsed / 100, and round the result to 1 decimal place.

If ***Calc Total Carbon Burned*** is greater than or equal to 0,
Add ***Calc Fuel Carbon Burned*** to ***Calc Total Carbon Burned.***

else
Set ***Calc Total Carbon Burned*** to -1.

if (***Current Daily Fuel Record***.FuelCarbonBurned is null)
return result A.

else if (***Current Daily Fuel Record***.FuelCarbonBurned is less than or equal to 0)
return result A.

else if (***Calc Fuel Carbon Burned*** is not null AND***Current Daily Fuel Record***.FuelCarbonBurned <> ***Calc Fuel Carbon Burned***)

Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "CARBON" AND
UOM = "LB"

if (ABS(***Current Daily Fuel Record***.FuelCarbonBurned - ***Calc Fuel Carbon Burned***) > *Tolerance*)
return result B.

Results:

Result	Response	Severity
A	The [fieldname] reported in the Daily Fuel record for [key] is missing or invalid. The value must be greater than 0.	Critical Error Level 1
B	The [fieldname] in the Daily Fuel record for [key] is inconsistent with the recalculated value.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2 Daily Emissions Fuel

Check Code: DAILY-12

Check Name: Intialize Daily Emissions

Related Former Checks:

Applicability:

Description:

Specifications:

Set ***Calc Total Carbon Burned*** to 0.
Daily Op Time Accumulator Array for the location = 0.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- CO2 Daily Emissions Initialization

Check Code: DAILY-13

Check Name: Check Total Carbon Burned

Related Former Checks:

Applicability:

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

Set ***Calculate CO2M TDE*** to true.

if (***Current CO2 Mass Daily Record***.TotalCarbonBurned is null)
    if (***Calc Total Carbon Burned*** is not equal to 0)
        return result A.

else if (***Current CO2 Mass Daily Record***.TotalCarbonBurned is less than 0)
    return result B.

else
    if (***Calc Total Carbon Burned*** is greater than 0 AND ***Current CO2 Mass Daily Record***.TotalCarbonBurned <> ***Calc Total Carbon Burned***)

        Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
            Parameter = "CARBON" AND
            UOM = "LB"

        if (ABS(***Current CO2 Mass Daily Record***.TotalCarbonBurned - ***Calc Total Carbon Burned***) > *Tolerance*)
            return result C.

    else if (***Calc Total Carbon Burned*** == 0)
        Set ***Calc Total Carbon Burned*** to ***Current CO2 Mass Daily Record***.TotalCarbonBurned.

Results:

Result	Response	Severity
A	You did not report TotalCarbonBurned in the Daily Emission record for CO2M. You must report this value if you report Daily Fuel records.	Critical Error Level 1
B	The [fieldname] reported in the Daily Emissions record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
C	The [fieldname] reported in the Daily Emissions record for [param] is inconsistent with the recalculated value.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2 Daily Emissions

Check Category:

Daily Interference Status

Check Code: INTSTAT-1

Check Name: Determine the Online Daily Interference Check

Related Former Checks:

Applicability: CEM Check

Description: This check locates the most recent prior daily interference check for the FLOW monitor used during the current hour.

Specifications:

Set ***OnlineDailyIntCheck*** = null
Set ***OnlineDailyIntRecord*** = null
Set ***OfflineDailyIntRecord*** = null.
Set ***DailyIntStatusResult*** = null

Locate ***OnlineDailyIntCheck*** in ***LatestDailyInterferenceCheckObject*** for the location where:
a) ComponentID is equal to the ***QaStatusComponentId*** AND
b) Online equals true.

if (***OnlineDailyIntCheck*** is not null)

Set ***OnlineDailyIntRecord*** = ***OnlineDailyIntCheck*** .DailyInterferenceCheckRow

If (***OnlineDailyIntRecord***.TestResultCd = "PASSED")

If (the number of clock hours between the ***OnlineDailyIntRecord***.EndDate/Hour and the ***CurrentMHVRecord***.Date/Hour is less than 26)
Set ***DailyIntStatusResult*** = "IC"

else if (***OnlineDailyIntRecord***.TestResultCd = "FAILED")
Set ***DailyIntStatusResult*** = "OOC-Test Failed"

else if (***OnlineDailyIntRecord***.TestResultCd = "ABORTED")
Set ***DailyIntStatusResult*** = "OOC-Test Aborted"

else
Set ***DailyIntStatusResult*** = "OOC-Test Has Critical Errors"

If (***DailyIntStatusResult*** is not equal to "IC")

Locate the latest record in ***DailyIntCheckRecordsByLocationForQAStatus*** for the location where:

a) Date/Hour is on or prior to the ***Current MHV Record.***Date/Hour
b) Date/Hour/Min is after the ***OnlineDailyIntRecord***.EndDate/Hour/Min
c) The ComponentID is equal to the ***QaStatusComponentId*** AND
d) TestResultCd is equal to "IGNORED"

if (***DailyIntCheckRecordsByLocationForQAStatus*** is found

Set ***OfflineDailyIntRecord*** = the found record in ***DailyIntCheckRecordsByLocationForQAStatus***.

If (***DailyIntStatusResult*** is not null)
Set ***DailyIntStatusResult*** = ***DailyIntStatusResult*** & "*"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Daily Interference Check Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Daily Interference Status Evaluation

Check Code: INTSTAT-2

Check Name: Determine Daily Interference Status for No Prior Check

Related Former Checks:

Applicability: CEM Check

Description: This check determines the daily inteference check status when there is no prior check.

Specifications:

If (***DailyIntStatusResult*** is null) and (***OnlineDailyIntRecord*** is null)

// Determine whether check in previous quarter is possibly effective for current hour.
If (the number of clock hours between the ***First Day of Operation/First Hour of Operation*** and the ***CurrentMHVRecord***.Date/Hour is less than 25)

Set ***DailyIntStatusResult*** = "IC-Undetermined".

else
// If a non operating hour exists within the first 24 hours after the first operating hour in the quarter, a grace period exists for seven hours after the operating hour subsequent to the non operating hour.
Locate the latest record in ***HourlyOpData*** where:

a) Date/Hour is ON OR PRIOR to the 24th clock hour following the ***First Day of Operation/First Hour of Operation***
b) OpTime is equal to zero.

if (***HourlyOpData*** is found)

Locate the first record in ***HourlyOpData*** where:

a) Date/Hour is after the Date/Hour in the ***HourlyOpData*** record found above
b) Date/Hour is ON OR PRIOR to the ***CurentMHVRecord***.Date/Hour
c) OpTime is greater than zero.

if (***HourlyOpData*** is found) and (the number of clock hours starting at ***HourlyOpData***.Date/Hour and up to the hour before ***CurrentMHVRecord***.Date/Hour is greater than 7)

Set ***DailyIntStatusResult*** = "OOC-No Prior Test".

else
Set ***DailyIntStatusResult*** = "IC-Undetermined".

else

Set ***DailyIntStatusResult*** = "OOC-No Prior Test".

If (***DailyIntStatusResult*** begins with "OOC")

Locate the record in ***DailyIntCheckRecordsByLocationForQAStatus*** for the location where:

a) Date/Hour is on or prior to the ***Current MHV Record.***Date/Hour
b) the ComponentID is equal to the ***QaStatusComponentId*** AND
c) TestResultCd is equal to "IGNORED"

if (***DailyIntCheckRecordsByLocationForQAStatus*** is found)

Set ***OfflineDailyIntRecord*** = the found record in ***DailyIntCheckRecordsByLocationForQAStatus***.
Set ***DailyIntStatusResult*** = ***DailyIntStatusResult*** & "*"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Daily Interference Check Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Daily Interference Status Evaluation

Check Code: INTSTAT-3

Check Name: Determine Expiration Status for Prior Daily Interference Check

Related Former Checks:

Applicability: CEM Check

Description: This check determines the daily interference status if the prior check is more than 26 hours prior to current hour.

Specifications:

If (***DailyIntStatusResult*** is null)

If (***OnlineDailyIntCheck***.LastCoveredNonOpHour is NOT null)

// If a non operating hour exists within the first 27 hours after the hour of the online Daily Interference Check, a grace period exists for eight hours starting with the operating hour subsequent to the non operating hour.
If (***OnlineDailyIntCheck***.FirstOpHourAfterLastNonOpHour is NOT null) AND (the number of clock hours inclusively between the ***OnlineDailyIntCheck***.FirstOpHourAfterLastNonOpHour and the ***CurrentDateHour*** is greater than 8)

Set ***DailyIntStatusResult*** = "OOC-Expired".

else

Set ***DailyIntStatusResult*** = "IC-Grace".

else

Set ***DailyIntStatusResult*** = "OOC-Expired".

If (***DailyIntStatusResult*** begins with "OOC" and ***OfflineDailyIntRecord*** is not null)
Set ***DailyIntStatusResult*** = ***DailyIntStatusResult*** & "*"

If (***DailyIntStatusResult*** does not begin with "IC")

Return ***DailyIntStatusResult*** .

Results:

Result	Response	Severity
OOC-Expired	The prior daily interference check for [compkey] completed on [date] expired.	Critical Error Level 1
OOC-Expired*	The prior daily interference check for [compkey] completed on [date] expired. A daily interference check completed on [invdate] was ignored because it was completed while the unit was offline.	Critical Error Level 1
OOC-No Prior Test	You did not report a prior daily interference check for [compkey] during the reporting period. Any daily interference check that may have been completed in a prior reporting period has expired.	Critical Error Level 1
OOC-No Prior Test*	You did not report a prior daily interference check for [compkey] during the reporting period. Any daily interference check that may have been completed in a prior reporting period has expired. A daily interference check completed on [invdate] was ignored because it was completed while the unit was offline.	Critical Error Level 1
OOC-Test Aborted	The prior daily interference check for [compkey] completed on [date] was aborted.	Critical Error Level 1
OOC-Test Aborted*	The prior daily interference check for [compkey] completed on [date] was aborted. An daily interference check completed on [invdate] was ignored because it was performed while the unit was offline.	Critical Error Level 1
OOC-Test Failed	The prior daily interference check for [compkey] completed on [date] failed.	Critical Error Level 1
OOC-Test Failed*	The prior daily interference check for [compkey] completed on [date] failed. An daily interference check completed on [invdate] was ignored because it was performed while the unit was offline.	Critical Error Level 1
OOC-Test Has Critical Errors	The prior daily interference check for [compkey] completed on [date] has critical errors.	Critical Error Level 1
OOC-Test Has Critical Errors*	The prior daily interference check for [compkey] completed on [date] has critical errors. An daily interference check completed on [invdate] was ignored because it was performed while the unit was offline.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Daily Interference Check Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Daily Interference Status Evaluation

Check Category:

Daily Test

Check Code: EMTEST-1

Check Name: Daily Test Date Valid

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

For the daily emission test:

Set ***EM Test Date Valid*** to true.

If Date is null,
set ***EM Test Date Valid*** to false, and return result A.

If Date is before 01/01/1993 or after the end of the ***Current Reporting Period***,
set ***EM Test Date Valid*** to false, and return result B.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Fatal
B	You reported a [Fieldname] of [Date], which is outside the range of acceptable values for this date for [key].	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test
2	Process/Category:	Emissions Data Evaluation Report ------ Miscellaneous Emission File Test

Check Code: EMTEST-2

Check Name: Daily Test Hour Valid

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

For the daily emission test:

Set ***EM Test Hour Valid*** to true.

If Hour is null,
Set ***EM Test Hour Valid*** to false, and return result A.

If Hour is not between 0 and 23,
Set ***EM Test Hour Valid*** to false, and return result B.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Fatal
B	You reported a [Fieldname] of [Hour], which is outside the range of acceptable values for this hour for [key].	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test
2	Process/Category:	Emissions Data Evaluation Report ------ Miscellaneous Emission File Test

Check Code: EMTEST-3

Check Name: Daily Test Minute Valid

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

For the daily emission test:

Set ***EM Test Minute Valid*** to true.

If Minute is null,

If (***Legacy Data Evaluation*** == false)
set ***EM Test Minute Valid*** to false, and return result A.

Otherwise,
return result B.

If Minute is not between 0 and 59,
set ***EM Test Minute Valid*** to false, and return result C.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Fatal
B	You did not provide [fieldname] for [key]. This information will be required for ECMPS submissions.	Informational Message
C	You reported a [Fieldname] of [Minute] for [key], which is outside the range of acceptable values.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test
2	Process/Category:	Emissions Data Evaluation Report ------ Miscellaneous Emission File Test

Check Code: EMTEST-4

Check Name: Daily Test System or Component Valid

Related Former Checks:

Applicability: General Check

Description:

Specifications:

For the daily test:

If both the MonitoringSystemID and ComponentID are not null,
return result A.

If TestTypeCode is equal to "INTCHK",

If ComponentID is null,
return result B.

If the ComponentTypeCode of the associated component is not equal to "FLOW",
return result C.

If TestTypeCode is equal to "PEMSCAL",

If MonitoringSystemID is null,
return result D.

If the SystemTypeCode of the associated system is not equal to "NOXP",
return result E.

Results:

Result	Response	Severity
A	You have reported both a MonitoringSystemID and a ComponentID for [key]. This is invalid.	Critical Error Level 1
B	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
C	The ComponentTypeCode for [key] is not appropriate for this type of test.	Critical Error Level 1
D	You did not provide a MonitoringSystemID for [key], which is required for this test type.	Critical Error Level 1
E	The SystemTypeCode of the system for [key] is not appropriate for this type of test.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Daily Calibration Test
2	Process/Category:	Emissions Data Evaluation Report ------ Miscellaneous Emission File Test

Check Code: EMTEST-5

Check Name: Daily Test Span Scale Valid

Related Former Checks:

Applicability: CEM Check

Description: This check determines whether the reported span scale is valid and consistent with the current analyzer range of the component.

Specifications:

For the daily test:

If the SpanScaleCode is not null,
return result A.

Results:

Result	Response	Severity
A	You reported [fieldname] for [key], which is not appropriate for this test type.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Miscellaneous Emission File Test

Check Code: EMTEST-6

Check Name: Daily Test Result Code Valid

Related Former Checks:

Applicability: General Check

Description:

Specifications:

For the daily test:

Set ***EM Test Calc Result*** to null.

If TestResultCode is null,
return result A.

else if TestResultCode is not equal to "ABORTED", "PASSED", or "FAILED",
return result B.

else
Set ***EM Test Calc Result*** to TestResultCode.

if TestTypeCode is equal to "INTCHK" and ***EM Test Date Valid*** and ***EM Test Hour Valid*** and OpTime is equal to 0,

Set ***Ignored Daily Interference Tests*** to true.
Set ***EM Test Calc Result*** to "IGNORED".

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	You reported the value [value], which is not in the list of valid values for this test type, in the field [fieldname] for [key].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Miscellaneous Emission File Test

Check Category:

EM Weekly System Integrity Test

Check Code: EMWSI-1

Check Name: Check Hg Converter Indicator of the Component

Related Former Checks:

Applicability:

Description: Ensures that the Hg Convert Indicator of the component associated with the test is set to 1.

Specifications:

If (***CurrentWeeklySystemIntegrityTest*** .HgConverterIndicator is NOT equal to 1)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Results:

Result	Response	Severity
A	For [key] you reported a HgConverterIndicator that is not equal to 1, which indicates that a Weekly System Integrity Test is not necessary.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWSI-2

Check Name: Check Gas Level

Related Former Checks:

Applicability:

Description: Ensure that the Gas Level was reported and with a valid value.

Specifications:

For ***CurrentWeeklySystemIntegrityTest***

If (GasLevelCode is null)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Else if (GasLevelCode is NOT in set (HIGH, MID, LOW, ZERO))

Set ***WeeklyTestSummaryValid*** to false.
return result B.

Else if (GasLevelCode is NOT in set (HIGH, MID))

Set ***WeeklyTestSummaryValid*** to false.
return result C.

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1
B	For [key] you reported a [levelcode] that is not in the list of valid [fieldname] for this test type.	Critical Error Level 1
C	For [key], you reported an invalid Gas Level Code of [levelcode], for a [testype].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWSI-3

Check Name: Check Weekly System Integrity Reference Value

Related Former Checks:

Applicability:

Description: Ensure that the Weekly System Integrity Test Reference Value was reported and with a valid value.

Specifications:

For ***CurrentWeeklySystemIntegrityTest***

Set ***InjectionReferenceValueValid*** to false.

If (ReferenceValue is null)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Else if (ReferenceValue is NOT rounded to one decimal place)

Set ***WeeklyTestSummaryValid*** to false.
return result B.

Else if (ReferenceValue is NOT greater than 0)

If (TestResultCode is NOT equal to "FAILED")

Set ***WeeklyTestSummaryValid*** to false.
return result C.

Else

Set ***InjectionReferenceValueValid*** to true

.

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1
B	The [fieldname] value for [key] should be reported to one decimal place.	Critical Error Level 1
C	Your reported CEM Value and/or Reference Value for [key] is less than or equal to zero.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWSI-4

Check Name: Check Weekly System Integrity Measured Value

Related Former Checks:

Applicability:

Description: Ensure that the Weekly System Integrity Test Measured Value was reported and with a valid value.

Specifications:

For ***CurrentWeeklySystemIntegrityTest***

Set ***InjectionMeasuredValueValid*** to false.

If (MeasuredValue is null)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Else if (MeasuredValue is NOT rounded to one decimal place)

Set ***WeeklyTestSummaryValid*** to false.
return result B.

Else if (MeasuredValue is NOT greater than 0)

If (TestResultCode is NOT equal to "FAILED")

Set ***WeeklyTestSummaryValid*** to false.
return result C.

Else

Set ***InjectionMeasuredValueValid*** to true

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1
B	The [fieldname] value for [key] should be reported to one decimal place.	Critical Error Level 1
C	Your reported CEM Value and/or Reference Value for [key] is less than or equal to zero.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWSI-5

Check Name: Calculate System Integrity Error and Alternate Performance Spec Indicator

Related Former Checks:

Applicability:

Description: Use the Reference and Measured Values to calculate the System Integrity Error and the Alternate Performance Spec. Also updates the calculated Weekly Test Summary Test Result.

Specifications:

For ***CurrentWeeklySystemIntegrityTest***

Set ***CalculatedSystemIntegrityApsIndicator*** to null.
Set ***CalculatedSystemIntegrityError*** to null.

If (***InjectionReferenceValueValid*** AND ***InjectionMeasuredValueValid***)

Set *PercentError* to (100 * ABS(ReferenceValue - MeasuredValue) / ReferenceValue), rounded to 1 decimal place.

If (*PercentError* is less than or equal to 10)

Set ***CalculatedSystemIntegrityApsIndicator*** to 0.
Set ***CalculatedSystemIntegrityError*** to *PercentError*.
Set ***CalculatedWeeklyTestSummaryResult*** to "PASSED".

Else
Set *AbsoluteError* to ABS(ReferenceValue - MeasuredValue), rounded to 2 decimal places.

If (*AbsoluteError* is less than or equal to 0.8)

Set ***CalculatedSystemIntegrityApsIndicator*** to 1.
Set ***CalculatedSystemIntegrityError*** to *AbsoluteError*.
Set ***CalculatedWeeklyTestSummaryResult*** to "PASSAPS".

Else

Set ***CalculatedSystemIntegrityApsIndicator*** to 0.
Set ***CalculatedSystemIntegrityError*** to *PercentError*.
Set ***CalculatedWeeklyTestSummaryResult*** to "FAILED".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWSI-6

Check Name: Check Weekly System Integrity Alternative Performance Spec

Related Former Checks:

Applicability:

Description: Ensures that the APS indicator is a valid value and matches the calculate APS indicator.

Specifications:

For ***CurrentWeeklySystemIntegrityTest***

Set ***WeeklySystemIntegrityApsIsValid*** to false.

If (***InjectionReferenceValueValid*** and ***InjectionMeasuredValueValid***)

If (ApsIndicator is null)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Else if (ApsIndicator is NOT equal to 0 OR 1)

Set ***WeeklyTestSummaryValid*** to false.
return result B.

Else if (ApsIndicator is NOT equal to ***CalculatedSystemIntegrityApsIndicator***)

Set ***WeeklyTestSummaryValid*** to false.
return result C.

Else
Set ***WeeklySystemIntegrityApsIsValid*** to true.

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1
B	You did not report an APSIndicator of "0" or "1" for [key].	Critical Error Level 1
C	The APSIndicator reported for [key] is inconsistent with the APSIndicator recalculated from the reported reference and measured values.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWSI-7

Check Name: Check Weekly System Integrity Error

Related Former Checks:

Applicability:

Description: Ensure that the Weekly System Integrity Error was reported, has a valid value, and matches the calculated value.

Specifications:

For ***CurrentWeeklySystemIntegrityTest***

Set ***WeeklySystemIntegrityErrorIsValid*** to false.

If (***InjectionReferenceValueValid*** and ***InjectionMeasuredValueValid***)

If (SystemIntegrityError is null)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Else if (***WeeklySystemIntegrityApsIsValid***)

If (SystemIntegrityError is NOT rounded to one decimal place)

Set ***WeeklyTestSummaryValid*** to false.
return result B.

Else if (SystemIntegrityError is NOT equal to ***CalculatedSystemIntegrityError***)

Set ***WeeklyTestSummaryValid*** to false.
return result C.

Else

Set ***WeeklySystemIntegrityErrorIsValid*** to true.

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1
B	The [testtype] status for [key] could not be determined, because the OperatingTime in at least one Hourly Operating Data records was missing or invalid.	Critical Error Level 1
C	The [fieldname] value for [key] is inconsistent with the value of [value], recalculated from the reported [testtype] records.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWSI-8

Check Name: Check Weekly Test Summary Result Against Calculated Value

Related Former Checks:

Applicability:

Description: Compares the calculated test result to the reported test result.

Specifications:

If (***CalculatedWeeklyTestSummaryResult*** is NOT null) and (***CurrentWeeklyTestSummary***.TestResultCode is NOT equal to ***CalculatedWeeklyTestSummaryResult***)

Set ***CalculatedWeeklyTestSummaryResult*** to null.
Set ***WeeklyTestSummaryValid*** to false.
return result A.

Results:

Result	Response	Severity
A	For [key], the [fieldname] is not consistent with the test result recalculated from the reported records.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWSI-9

Check Name: Update Weekly System Integrity Dictionary Component Entry

Related Former Checks:

Applicability:

Description: Initializes a WsiTestDictionary entry for the CurrentWeeklySystemIntegrityTest Component Id if it does not already exist.

If the LastEvaluatedTestRecord is not null and its TestDateHour is before the current hour, assign it as the MostRecentTestRecord, and clear the operating date list and last operating date.

Finally always set the LastEvaluatedTestRecord to the CurrentWeeklySystemIntegrityTest.

Specifications:

If (***WsiTestDictionary*** does NOT contain a key equal to ***CurrentWeeklySystemIntegrityTest***.ComponentId)

Add an entry to ***WsiTestDictionary*** for ***CurrentWeeklySystemIntegrityTest***.ComponentId with the following fields:

a) MostRecentTestRecord
b) OperatingDateList initialized as an empty list.

For the ***WsiTestDictionary*** entry where the key is equal to ***CurrentWeeklySystemIntegrityTest***.ComponentId:

Set MostRecentTestRecord to ***CurrentWeeklySystemIntegrityTest***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWSI-10

Check Name: Update Weekly System Integrity Dictionary Operating Date Information

Related Former Checks:

Applicability:

Description: Updates the OperatingDateList with the current date if it is not in the list and the current hour is an operating hour.

Specifications:

If (***CurrentOperatingTime*** is greater than 0)

For each entry in ***WsiTestDictionary***:

If (MostRecentTestRecord is NOT null) AND (MostRecentTestRecord .LocationKey is equal to ***CurrentMonitorLocationId***)

If (MostRecentTestRecord .TestDateHour is equal to ***CurrentDateHour***)

Set OperatingDateList to an empty list.

If (MostRecentTestRecord.TestDate is prior to ***CurrentDateHour***) AND (OperatingDateList does NOT contain ***CurrentOperatingDate***)

Add ***CurrentOperatingDate*** to OperatingDateList

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Weekly System Integrity Test Operating Dates

Check Code: EMWSI-11

Check Name: Ensure that Weekly System Integrity Test Occurred During an Operating Hour

Related Former Checks:

Applicability: General Check

Description: This check ensures that the Weekly System Integrity test occurred during an operating hour.

Specifications:

For ***CurrentWeeklySystemIntegrityTest***

If (***CurrentOperatingTime*** is equal to 0)

Return result A

Results:

Result	Response	Severity
A	Test [key] was performed while the unit was not operating even though [type] test are only allowed during operating hours.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Category:

EM Weekly Test Summary

Check Code: EMWTS-1

Check Name: Initialize Parameters

Related Former Checks:

Applicability:

Description: Initializes the updatable parameters used in weekly test summary evaluations.

Specifications:

Set ***WeeklyTestSummaryValid*** to true.
Set ***CalculatedWeeklyTestSummaryResult*** to null.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWTS-2

Check Name: Check Weekly Test Type

Related Former Checks:

Applicability:

Description: Ensures that the test type is for a weekly test.

Specifications:

For ***CurrentWeeklyTestSummary***

If (TestTypeCode is not equal to "HGSI1"),

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Results:

Result	Response	Severity
A	You reported a [testtype] in [key] that is not a valid TestTypeCode for a weekly test.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWTS-3

Check Name: Check Weekly Test System

Related Former Checks:

Applicability:

Description: Ensures that the system identifer is either reported or not reported when expected, and if it is and should have been reporte, that it was reported with the correct system type.

Specifications:

For ***CurrentWeeklyTestSummary***

If (TestTypeCode is equal to "HGSI1"),

If (SystemId is NOT null),

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Results:

Result	Response	Severity
A	You reported a MonitoringSystemID for [key], which is not valid for a [testtype]. Only a ComponentID is reported for a [testtype].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWTS-4

Check Name: Check Weekly Test Component

Related Former Checks:

Applicability:

Description: Ensures that the system identifer is either reported or not reported when expected, and if it is and should have been reporte, that it was reported with the correct system type.

Specifications:

For ***CurrentWeeklyTestSummary***

If (TestTypeCode is equal to "HGSI1")

If (ComponentId is null)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Else if (ComponentTypeCode is NOT equal to "HG")

Set ***WeeklyTestSummaryValid*** to false.
return result B.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	The ComponentTypeCode for [key] is not appropriate for this type of test.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWTS-5

Check Name: Check Weekly Test Date

Related Former Checks:

Applicability:

Description: Ensures that an date was reported for the weekly test and that the value reported is valid.

Specifications:

For ***CurrentWeeklyTestSummary***

Set ***TestDateValid*** to false.

If (TestDate is null)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Else if (TestDate is before 01/01/1993) OR (TestDate is after ***CurrentReportingPeriodEndHour***)

Set ***WeeklyTestSummaryValid*** to false.
return result B.

Else
Set ***TestDateValid*** to true.

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1
B	You reported a [Fieldname] of [Date], which is outside the range of acceptable values for this date for [key].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWTS-6

Check Name: Check Weekly Test Hour

Related Former Checks:

Applicability:

Description: Ensures that an hour was reported for the weekly test and that the value reported is valid.

Specifications:

For ***CurrentWeeklyTestSummary***

Set ***TestHourValid*** to false.

If (TestHour is null)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Else if (TestHour is NOT between 0 and 23)

Set ***WeeklyTestSummaryValid*** to false.
return result B.

Else
Set ***TestHourValid*** to ***TestDateValid***.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Critical Error Level 1
B	You reported a [Fieldname] of [Hour], which is outside the range of acceptable values for this hour for [key].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWTS-7

Check Name: Check Weekly Test Minute

Related Former Checks:

Applicability:

Description: Ensures that a minute was reported for the weekly test and that the value reported is valid.

Specifications:

For ***CurrentWeeklyTestSummary***

Set ***TestDateTimeValid*** to false.

If (TestMinute is null)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Else if (TestMinute is NOT between 0 and 59)

Set ***WeeklyTestSummaryValid*** to false.
return result B.

Else if (***TestHourValid)***

Set ***TestDateTimeValid*** to true.

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1
B	You reported a [Fieldname] of [Minute] for [key], which is outside the range of acceptable values.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWTS-8

Check Name: Check Weekly Test Span Scale

Related Former Checks:

Applicability:

Description: Ensure that the Weekly Test Gas Level was reported and with a valid value.

Specifications:

For ***CurrentWeeklyTestSummary***

If (SpanScaleCode is null)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Else if (SpanScaleCode is NOT in set (H, M, L))

Set ***WeeklyTestSummaryValid*** to false.
return result B.

Else if (TestTypeCode is equal to "HGSI1")

If (SpanScaleCode is NOT equal to "H")

Set ***WeeklyTestSummaryValid*** to false.
return result C.

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1
B	For [key], you reported a SpanScaleCode that in not an appropriate code for a [testtype].	Critical Error Level 1
C	For [key], you reported a SpanScaleCode that in not an appropriate code for a [testtype].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Code: EMWTS-9

Check Name: Check Weekly Test Result

Related Former Checks:

Applicability:

Description: Ensure that the Weekly Test Result was reported and with a valid value.

Specifications:

For ***CurrentWeeklyTestSummary***

Set ***TestResultValid*** = false.

If (TestResultCode is null)

Set ***WeeklyTestSummaryValid*** to false.
return result A.

Else if (TestResultCode is not in ***TestResultCodeList***)

Set ***WeeklyTestSummaryValid*** to false.
return result B.

Else if (TestTypeCode is equal to "HGSI1")

If (TestResultCode is NOT in set (PASSED, PASSAPS, FAILED))

Set ***WeeklyTestSummaryValid*** to false.
return result C.

Else
Set ***TestResultValid*** = true.

Else
Set ***TestResultValid*** = true.

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1
B	You reported the value [value], which is not in the list of valid values for this test type, in the field [fieldname] for [key].	Critical Error Level 1
C	You reported the value [value], which is not in the list of valid values for this test type, in the field [fieldname] for [key].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Weekly System Integrity Test Evaluation

Check Category:

Emissions Audit Checks

Check Code: EMAUDIT-1

Check Name: Link-Kind Use Audit

Related Former Checks:

Applicability:

Description: Returns a result if a Link-Kind monitor was used during the quarter. The result is Informational if the monitor was used for 720 hours or less during the calendar year and Critical 1 if it was used for more than 720 hours during the year.

Specifications:

Set ***LikeKindHours*** to null.

When ***ComponentRecordForAudit***.ComponentIdentifier begins with "LK":

Set *LocationPosition* to entry in ***LocationPositionLookup*** for ***ComponentRecordForAudit***.MonLocId.

Set *DictionaryEntry* to entry in ***ComponentOperatingSuppDataDictionaryArray*** for *LocationPosition* AND ***ComponentRecordForAudit***.ComponentId where OpSuppDataType equals "OP".

If (*DictionaryEntry* exists) AND (*DictionaryEntry*.Hours is greater than 0)

Set ***LikeKindHours*** to *DictionaryEntry*.Hours.

If (***CurrentReportingPeriodObject***.Quarter is greater than 1)

Locate *SupplementalRecords* in ***ComponentOperatingSuppDataRecordsForMpAndYear*** where:

1) ComponentId equals ***ComponentRecordForAudit***.ComponentId.
2) Quarter is before ***CurrentReportingPeriodObject***.Quarter
3) OpSuppDataType equals "OP".

For each *SupplementalRecord* in *SupplementalRecords*.

Add *SupplementalRecord*.Hours to ***LikeKindHours*** .

If (***LikeKindHours*** is greater than 720)

Return result A.

Else

Return result B.

Results:

Result	Response	Severity
A	You have reported hourly data using Like Kind component '[ComponentIdentifier]' for [Hours] hours this year. [Hours] hours exceed the ANNUAL limit of 720 hours.	Critical Error Level 1
B	You have reported hourly data using Like Kind component '[ComponentIdentifier]' for [Hours] hour(s) this year. Please be aware that the use of like kind monitors is limited to less than 720 hours on an annual basis.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report --- Component Audit

Check Category:

Flow-to-Load Status

Check Code: F2LSTAT-1

Check Name: Determine Most Recent Flow-to-Load QA Operating Quarter

Related Former Checks:

Applicability:

Description: This check finds the most recent QA operating quarter for flow-to-load checks.

Specifications:

Set ***F2LStatusPriorTestRequiredQuarter*** = null.
Set ***F2LStatusPriorTestRequiredQuarterMissingOpData*** = null.

If ***F2LStatusSystemResultDictionary*** does not contain lookup value for ***CurrentMhvRecord***.SystemID

Locate the most recent record in ***RataTestRecordsByLocationForQaStatus*** where the SystemID is equal to ***CurrentMhvRecord***.SystemID, EndDateHour is before the ***CurrentReportingPeriod***, and TestResultCode is not equal to "INVALID"

If found,

For each quarter before ***CurrentReportingPeriod*** beginning with the quarter immediately before ***CurrentReportingPeriod*** and going back to later of the quarter of the located ***RataTestRecordsByLocationForQaStatus*** and the quarter of the ***Earliest Location Report Date***

If ***AnnualReportingRequirement*** equals true, or the quarter being checked is 2 or 3

If ***AnnualReportingRequirement*** equals true, or the quarter being checked is 3

Locate the record in ***OperatingSuppDataRecordsByLocation*** where OpTypeCode is equal to "OPHOURS", FuelCode is null, and reporting period equals the quarter being checked

Else

Locate the record in ***OperatingSuppDataRecordsByLocation*** where OpTypeCode is equal to "OSHOURS", FuelCode is null, and reporting period equals the quarter being checked

If ***OperatingSuppDataRecordsByLocation*** record is found

If ***F2LStatusPriorTestRequiredQuarter*** does not equal -1

If ***OperatingSuppDataRecordsByLocation***.OpValue >= 168

Locate a record in ***F2LCheckRecordsForQaStatus*** where SystemID is equal to ***CurrentMhvRecord***.SystemID, the quarter is equal to the quarter being checked, and TestResultCode is equal to "EXC168H" or "FEW168H"

If not found

Set ***F2LStatusPriorTestRequiredQuarter***.Year = The year value of the quarter being checked.
Set ***F2LStatusPriorTestRequiredQuarter***.Quarter = The quarter value of the quarter being checked.
Exit the check.

Else

Set ***F2LStatusPriorTestRequiredQuarter*** = -1.
Append "[YEAR]Q[QTR]" to ***F2LStatusPriorTestRequiredQuarterMissingOpData*** (where

[YEAR] and [QTR are the year and number of the quarter being checked.

Else

Set ***F2LStatusPriorTestRequiredQuarter*** = -1.
Set ***F2LStatusPriorTestRequiredQuarterMissingOpData*** = "No Prior RATA"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Flow-to-Load Status Evaluation

Check Code: F2LSTAT-2

Check Name: Locate Most Recent Flow-to-Load Check Prior to the Current Hour

Related Former Checks:

Applicability:

Description: This check locates the most recent passed or failed flow-to-load (F2L) check for the emission report, insures that quarters between the quarter of the check and the emission report quarter were either not QA operating quarters (less than 168 operating hours) or have an F2L with a result of EXC168H, indicating that the operating hours minus exluded hours is less than 168.

If the check exist but an intervening quarter with more than 168 operating hours (excluding EXC168H) exists, or a check does not exist a parameter indicating the ultimate result is set. If the check does not exist and the a non load based unit is involved (directly or indirectly) or a Flow-to-Load exemption exist, then the result is IC-Exempt. If a RATA exists in the prior quarter then the result is OOC-No Prior Check, and if it does not then IC-No Prior RATA.

Specifications:

If ***F2LStatusSystemResultDictionary*** contains lookup value for ***CurrentMhvRecord***.SystemID

Set ***F2LStatusResult*** = ***F2LStatusSystemResultDictionary*** lookup value for ***CurrentMhvRecord***.SystemID.
Set ***CurrentFlowToLoadStatusCheck*** = ***F2LStatusSystemCheckDictionary*** lookup value for ***CurrentMhvRecord***.SystemID.
Set ***F2LStatusMissingOpDataInfo*** = ***F2lStatusSystemMissingOpDictionary*** lookup value for ***CurrentMhvRecord***.SystemID.

Else

Set ***F2LStatusResult*** = null.
Set ***CurrentFlowToLoadStatusCheck*** = null.
Set ***F2LStatusMissingOpDataInfo*** = null.

Locate the most recent record in ***F2LCheckRecordsForQaStatus*** where SystemID is equal to ***CurrentMhvRecord***.SystemID, EndDate < ***CurrentReportingPeriodBeginDateHour***, and TestResultCode is equal to "PASSED" or "FAILED"

If not found

Locate a record in ***MpLocationNonLoadBasedRecords*** where the location is the location in ***CurrentMhvRecord***.

If found, and NonLoadBaseInd equals 1

Set ***F2LStatusResult*** = "IC-Exempt".

Else

Locate a record in ***TestExtensionExemptionRecords*** where the SystemID is equal to the ***CurrentMhvRecord***.SystemID, the ExtensionExemptionCode is equal to "F2LEXP", and the reporting period is the period before the current reporting period.

If found

Set ***F2LStatusResult*** = "IC-Exempt".

Else

Locate the most recent record in ***RataTestRecordsByLocationForQaStatus*** where the SystemID is equal to ***CurrentMhvRecord***.SystemID, EndDate < ***CurrentReportingPeriodBeginDateHour***, and TestResultCode is not equal to "INVALID"

If not found

Set ***F2LStatusResult*** = "IC-No Prior RATA".

Else if ***F2LStatusPriorTestRequiredQuarter*** is equal to -1

Set ***F2LStatusResult*** = "Missing Op Data".
Set ***F2LStatusMissingOpDataInfo*** = ***F2LStatusPriotTestRequiredQuarterMissingOpData***.

Else if ***F2LStatusPriorTestRequiredQuarter*** is null or before the quarter of the located ***RataTestRecordsByLocationForQaStatus*** record

Set ***F2LStatusResult*** = "IC".

Else if the quarter of the located ***RataTestRecordsByLocationForQaStatus*** record is the quarter before ***CurrentReportingPeriod***, the TestReasonCode equals "INITIAL" or "RECERT" and TestResultCode equals "PASSED"

Set ***F2LStatusResult*** = "IC".

Else if ***CurrentMhvRecord***.SystemDesignationCode is equal to "RB"

Set ***F2LStatusResult*** = "Undetermined-No Prior Check reported for Redundant Backup Monitor".

Else

Set ***F2LStatusResult*** = "OOC-Prior Check Missing".

Else

Set ***CurrentFlowToLoadStatusCheck*** = the located ***F2LCheckRecordsForQaStatus*** record.

If the quarter of ***CurrentFlowToLoadStatusCheck*** is not the quarter before ***CurrentReportingPeriod***

If ***F2LStatusPriorTestRequiredQuarter*** is equal to -1

Set ***F2LStatusResult*** = "Missing Op Data".
Set ***F2LStatusMissingOpDataInfo*** = ***F2LStatusPriotTestRequiredQuarterMissingOpData***.

Else if ***F2LStatusPriorTestRequiredQuarter*** is not null, and is after the quarter of ***CurrentFlowToLoadStatusCheck***

if ***CurrentMhvRecord***.SystemDesignationCode is equal to "RB"

Set ***F2LStatusResult*** = "Undetermined-No Prior Check reported for Redundant Backup Monitor".

Else

Set ***F2LStatusResult*** = "OOC-Prior Check Missing".

Else if ***CurrentFlowToLoadStatusCheck***.TestResultCode = "PASSED"

Set ***F2LStatusResult*** = "IC".

Else

If ***CurrentFlowToLoadStatusCheck***.TestResultCode = "PASSED"

Set ***F2LStatusResult*** = "IC".

Set ***F2LStatusSystemResultDictionary*** lookup value for ***CurrentMhvRecord***.SystemID = ***F2LStatusResult***.
Set ***F2LStatusSystemCheckDictionary*** lookup value for ***CurrentMhvRecord***.SystemID = ***CurrentFlowToLoadStatusCheck***.
Set ***F2lStatusSystemMissingOpDictionary*** lookup value for ***CurrentMhvRecord***.SystemID = ***F2LStatusMissingOpDataInfo***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Flow-to-Load Status Evaluation

Check Code: F2LSTAT-3

Check Name: Locate Intervening RATA

Related Former Checks:

Applicability:

Description: This check locates RATA that occurred after the quarter of the (failed) most recent prior flow-to-load check and prior to the current hour.

Specifications:

Set ***F2LStatusInterveningRata*** = *null.*

If ***F2LStatusResult*** is null

Locate the most recent record in ***RataTestRecordsByLocationForQaStatus*** where the SystemID is equal to ***CurrentMhvRecord***.SystemID, EndDateHour is after ***CurrentFlowToLoadStatusCheck***.EndDateHour and before ***CurrentMhvRecord***.BeginDate/BeginHour, and TestResultCode is not equal to "INVALID"

If found,

Set ***F2LStatusResult*** = "IC-Subsequent RATA Performed".
Set ***F2LStatusInterveningRata*** = The located record in ***RataTestRecordsByLocationForQaStatus***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Flow-to-Load Status Evaluation

Check Code: F2LSTAT-4

Check Name: Locate Most Recent QA Cert Event

Related Former Checks:

Applicability:

Description: This check locates Abbreviated Flow-to-Load that occurred after the quarter of the (failed) most recent prior Flow-to-Load check and prior to the current hour.

Validation Tables:

[Test Type to Required Test Code] (Cross Check Table)

Specifications:

Set ***F2LStatusQaCertEvent*** = null.
Set ***F2L Status Event Requires RATA*** = false.
Set ***F2L Status Event Requires Abbreviated Check*** = false.

If ***F2LStatusResult*** is null

Locate the most recent record in ***QACertificationEventRecords*** where the SystemID is equal to ***CurrentMhvRecord***.SystemID, QaCertEventCode is equal to "312", QaCertEventDateHour is on or after ***CurrentFlowToLoadStatusCheck***.EndDateHour, and QaCertEventDateHour is on or before ***CurrentMhvRecord***.BeginDate/BeginHour.

If found,

Set ***F2LStatusQaCertEvent*** = The located record in ***QACertificationEventRecords***.

Locate a record in Cross-Check Table "Test Type to Required Test Code" where TestTypeCode begins with "RATA" and RequireTestCode equals ***F2LStatusQaCertEvent***.RequiredTestCode.

If found,

F2L Status Event Requires RATA = true.

Locate a record in Cross-Check Table "Test Type to Required Test Code" where TestTypeCode is equal to "AF2LCHK" and RequireTestCode equals ***F2LStatusQaCertEvent***.RequiredTestCode.

If found,

F2L Status Event Requires Abbreviated Check = true.

If ***F2LStatusQaCertEvent***.LastTestCompletedDateHour is on or before ***CurrentMhvRecord***.BeginDate/BeginHour

If ***F2L Status Event Requires Abbreviated Check*** is equal to true,

Set ***F2LStatusResult*** = "IC-Subsequent Abbreviated Flow-to-Load Check Passed".

If ***F2LStatusResult*** is null,

If ***F2LStatusQaCertEvent*** = null, or ***F2LStatusQaCertEvent***.ConditionalDataBeginDateHour is after ***CurrentMhvRecord***.BeginDate/BeginHour,

Locate records in ***RataTestRecordsByLocationForQaStatus*** where the SystemID is null or is equal to ***CurrentMhvRecord***.SystemID, EndDateHour is after ***CurrentFlowToLoadStatusCheck***.EndDateHour and before ***CurrentMhvRecord***.BeginDate/BeginHour, and TestResultCode is equal to "INVALID".

If found

Set ***F2LStatusResult*** = "OOC-Check Failed - Invalid RATA Ignored".

Else

Set ***F2LStatusResult*** = "OOC-Check Failed".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Flow-to-Load Status Evaluation

Check Code: F2LSTAT-5

Check Name: Locate Earliest Valid Required Test

Related Former Checks:

Applicability:

Description: This check locates the earliest RATA test if the most recent event had a Require Test Code of '5' or '6', or the earliest Abbreviated F2L if the Required Test Code is '26'. If the test was found and it failed the F2L status is OOC, but for RATA the OOC status depends on the existence of an intervening invalid RATA.

If the test does not exist or did not fail, the F2L is IC if the number of operating hours is less than or equal to 720 for RATA and 168 for other tests. Otherwise the status is OOC, with the RATA OOC value depending on whether an intevening invalid RATA exists.

Specifications:

Set ***F2lStatusEarliestValidRequiredTest*** = *null.*

If ***F2LStatusResult*** is null

If ***F2L Status Event Requires RATA*** is true,

Locate earliest record in ***RataTestRecordsByLocationForQaStatus*** where the SystemID is equal to ***CurrentMhvRecord***.SystemID, EndDateHour is after ***CurrentMhvRecord***.BeginDate/BeginHour, and TestResultCode is not equal to "INVALID".

If found,

Set ***F2LStatusEarliestValidRequiredTest*** = The located record in ***RataTestRecordsByLocationForQaStatus*** .

If ***F2LStatusEarliestValidRequiredTest*** .TestResultCode is equal to "FAILED"

Locate records in ***RataTestRecordsByLocationForQaStatus*** where the SystemID is equal to ***CurrentMhvRecord***.SystemID, EndDateHour is after ***F2LStatusQaCertEvent***.QaCertEventDateHour and before ***F2LStatusEarliestValidRequiredTest***.EndDateHour, and TestResultCode is equal to "INVALID".

If found

Set ***F2LStatusResult*** = "OOC-Recertification RATA Failed - Invalid RATA Ignored".

Else

Set ***F2LStatusResult*** = "OOC-Recertification RATA Failed".

Else if ***F2L Status Event Requires Abbreviated Check*** is false,

Set ***F2LStatusResult*** = "OOC-Invalid Cert Event".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Flow-to-Load Status Evaluation

Check Code: F2LSTAT-6

Check Name: Determine Event Conditional Status and Final Status

Related Former Checks:

Applicability:

Description: This check determines the status result when a QA Cert Event Conditional Data Period is involved.

Additionally, the check sets the final result for the Flow-to-Load Status checks.

Specifications:

If ***F2LStatusResult*** is null

Set ***F2L Status Missing Op Data Info*** = null.

If ***F2L Status Event Requires RATA*** is true,

Set *OperatingHourLimit* = 720

else

Set *OperatingHourLimit* = 168

If (the quarter of the ***F2LStatusQaCertEvent***.ConditionalBeginDate is equal to the quarter of the ***CurrentMhvRecord***.Date/Hour)

Count the number of ***HourlyOpData*** records for the location where OpTime is greater than 0 and Date/Hour is ON OR AFTER the ***F2LStatusQaCertEvent***.ConditionalBeginDate/Hour and ON OR BEFORE ***CurrentMhvRecord***.Date/Hour ,

If the number > *OperatingHourLimit* ,

Set ***F2LStatusResult*** = "OOC-Conditional Period Expired".

else

Set ***F2LStatusResult*** = "IC-Conditional".

else

if (***F2LStatusQaCertEvent***.MinOpHoursPriorQuarter is null)

Set ***F2LStatusQaCertEvent***.MinOpHoursPriorQuarter = 0
Set ***F2LStatusQaCertEvent***.MaxOpHoursPriorQuarter = 0

for each quarter beginning with the quarter of the ***F2LStatusQaCertEvent***.ConditionalBeginDate and continuing through the quarter BEFORE the ***CurrentMhvRecord***.Date/Hour :

if (***EarliestLocationReportDate*** <= the last day of the quarter being checked)

if (***Annual Reporting Requirement*** == false AND the quarter being checked == 2)

Locate the record in ***OperatingSuppDataRecordsbyLocation*** where the OpTypeCode is equal to "OSHOURS" and the reporting period is equal to the quarter being checked.

else

Locate the record in ***OperatingSuppDataRecordsbyLocation*** where the OpTypeCode is equal to "OPHOURS", FuelCode is null, and the reporting period is equal to the quarter

being checked.

if (***OperatingSuppDataRecordsbyLocation*** is not found)

Set ***F2LStatusQaCertEvent***.MinOpHoursPriorQuarter = -1
Append "[YEAR] Q[QTR]" to ***F2L Status Missing Op Data Info*** (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
exit for.

else

If the quarter being checked is the quarter of the ***F2LStatusQaCertEvent***.ConditionalBeginDate

If (***OperatingSuppDataRecordsbyLocation***.OpValue MINUS the number of calendar hours in the quarter being checked that are PRIOR to the ***F2LStatusQaCertEvent***.ConditionalBeginDate/Hour > 0)

Set ***F2LStatusQaCertEvent***.MinOpHoursPriorQuarter = ***OperatingSuppDataRecordsbyLocation***.OpValue MINUS the number of calendar hours in the quarter being checked that are PRIOR to the ***F2LStatusQaCertEvent***.ConditionalBeginDate/Hour

If (***OperatingSuppDataRecordsbyLocation***.OpValue is less than the number of calendar hours in the quarter begin checked that are ON OR AFTER the ***F2LStatusQaCertEvent***.ConditionalBeginDate/Hour)

Set ***F2LStatusQaCertEvent***.MaxOpHoursPriorQuarter = ***OperatingSuppDataRecordsbyLocation***.OpValue.

else

Set ***F2LStatusQaCertEvent***.MaxOpHoursPriorQuarter = the number of calendar hours in the quarter being checked that are ON OR AFTER the ***F2LStatusQaCertEvent***.ConditionalBeginDate/Hour.

else

Set ***F2LStatusQaCertEvent***.MinOpHoursPriorQuarter = ***F2LStatusQaCertEvent***.MinOpHoursPriorQuarter + ***OperatingSuppDataRecordsbyLocation***.OpValue.

Set ***F2LStatusQaCertEvent***.MaxOpHours PriorQuarter = ***F2LStatusQaCertEvent***.MaxOpHoursPriorQuarter + ***OperatingSuppDataRecordsbyLocation***.OpValue.

If (***F2LStatusQaCertEvent***.MinOpHoursPriorQuarter == -1)

set ***F2LStatusResult*** to "Missing Op Data"

Else if (***F2LStatusQaCertEvent***.MinOpHoursPriorQuarter > *OperatingHourLimit*)

If ***F2L Status Event Requires RATA*** is true,

Locate records in ***RataTestRecordsByLocationForQaStatus*** where the SystemID is equal to ***CurrentMhvRecord***.SystemID, EndDateHour is on or after

F2LStatusQaCertEvent.ConditionalDataBeginDate/ConditionalDataBeginHour and on or before ***CurrentMhvRecord***.BeginDate/BeginHour, and TestResultCode is equal to "INVALID".

If found,

Set ***F2LStatusResult*** = "OOC-Conditional Period Expired-Invalid RATA Ignored".

Else

Set ***F2LStatusResult*** = "OOC-Conditional Period Expired".

Else

Set ***F2LStatusResult*** = "OOC-Conditional Period Expired".

Else if (***Rpt Period Op Hours Accumulator Array*** for the location == -1)

Set ***F2LStatusResult*** = "Invalid Op Data".

Else if (***F2LStatusQaCertEvent***.MinOpHoursPriorQuarter + ***Rpt Period Op Hours Accumulator Array*** for the Location > *OperatingHourLimit*)

If ***F2L Status Event Requires RATA*** is true,

Locate records in ***RataTestRecordsByLocationForQaStatus*** where the SystemID is equal to ***CurrentMhvRecord***.SystemID, EndDateHour is on or after ***F2LStatusQaCertEvent***.ConditionalDataBeginDate/ConditionalDataBeginHour and on or before ***CurrentMhvRecord***.BeginDate/BeginHour, and TestResultCode is equal to "INVALID".

If found,

Set ***F2LStatusResult*** = "OOC-Conditional Period Expired-Invalid RATA Ignored".

Else

Set ***F2LStatusResult*** = "OOC-Conditional Period Expired".

Else

Set ***F2LStatusResult*** = "OOC-Conditional Period Expired".

Else if (***F2LStatusQaCertEvent***.MaxOpHoursPriorQuarter + ***Rpt Period Op Hours Accumulator Array*** for the Location > *OperatingHourLimit*)

Set ***F2LStatusResult*** = "Undetermined-Conditional Data".

Else

Set ***F2LStatusResult*** = "IC-Conditional".

If (***FlowToLoadStatusResult*** does not begin with "IC")

Return result ***F2LStatusResult*** .

Results:

Result	Response	Severity
Invalid Op Data	The Flow-to-Load status for [SYSID] could not be determined, because the OperatingTime in at least one Hourly Operating Data records was missing or invalid.	Critical Error Level 1
Missing Op Data	The Flow-to-Load status for [SYSID] could not be determined, because the Op Supp Data record for OPHOURS, OSHOURS, or OPDAYS is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to synchronize these records to your Client Tool by logging on to the EPA host.	Critical Error Level 1
OOC-Check Failed	The prior Flow-to-Load Check for SystemID [SYSID] has failed.	Critical Error Level 1
OOC-Check Failed - Invalid RATA Ignored	The prior Flow-to-Load Check for SystemID [SYSID] has failed, a subsequent invalid RATA with was ignored.	Critical Error Level 1
OOC-Conditional Period Expired	The conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [SYSID] has expired.	Critical Error Level 1
OOC-Conditional Period Expired-Invalid RATA Ignored	The conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for SystemID [EVENTKEY] has expired.	Critical Error Level 1
OOC-Invalid Cert Event	You reported an invalid QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for [SYSID].	Critical Error Level 1
OOC-Prior Check Missing	One or more Flow-to-Load Checks is missing for prior quarters.	Critical Error Level 1
OOC-Recertification RATA Failed	The subsequent recertification RATA for SystemID [SYSID] with TestNumber [subtestnum] failed.	Critical Error Level 1
OOC-Recertification RATA Failed - Invalid RATA Ignored	The subsequent recertification RATA for SystemID [SYSID] with TestNumber [subtestnum] failed. An invalid RATA was ignored.	Critical Error Level 1
Undetermined-Conditional Data	The software could not determine if the current hour was within the conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for SystemID [eventkey].	Informational Message
Undetermined-No Prior Check reported for Redundant Backup Monitor	The software could not determine if a Flow-to-Load check is required for the Redundant Backup Flow Monitor.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Flow-to-Load Status Evaluation

Check Category:

Hourly Aggregation

Check Code: HOURAGG-1

Check Name: Determine Start Quarter

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Set ***Start Quarter*** to null.
Set ***SO2 Start Quarter*** to null.
Set ***NOXR Start Quarter*** to null.
Set ***CO2 Start Quarter*** to null.
Set ***Heat Input Start Quarter*** to null.
Set ***NOX Start Quarter*** to null.
Set ***Emissions Tolerance Deviators*** to null.

If (Quarter of the ***Current Reporting Period*** is greater than 1)

Locate the earliest ***Monitor Method*** for location where BeginDate is on or before the last day of the ***Current Reporting Period***, and EndDate is null or is on or after Jan 1 of the year of the ***Current Reporting Period***.

If found,
If BeginDate is in a year prior to the current reporting period,
If (***Annual Reporting Requirement*** == true)
Set ***Start Quarter*** to 1.
else
Set ***Start Quarter*** to 2.
else
Set ***Start Quarter*** to the quarter of the BeginDate.

Locate the earliest ***Monitor Method*** for location where ParameterCode = "SO2" or "SO2M", BeginDate is on or before the last day of the ***Current Reporting Period***, and EndDate is null or is on or after Jan 1 of the year of the ***Current Reporting Period***.

If found,
If BeginDate is in a year prior to the current reporting period,
Set ***SO2 Start Quarter*** to 1.
else
Set ***SO2 Start Quarter*** to the quarter of the BeginDate.

If (***LME Annual*** == true)

Locate the record for the location with the earliest Quarter in ***NOXR Summary Required for LME Annual Records*** where LmeNoxrSummaryIndicator is equal to 1.

if found,
Set *EarliestMethodBeginDate* to the date from LmeNoxrBegin for the located record in ***NOXR Summary Required for LME Annual Records***.
else
Set *EarliestMethodBeginDate* to null.

else
if (***Current Monitor Plan Location Record***.LocationName begins with "MS" OR ***Multiple Stack Configuration*** == false)

Locate the earliest ***Monitor Method*** for location where ParameterCode = "NOXR", BeginDate is on or before the last day of the ***Current Reporting Period***, and EndDate is null or is on or after Jan 1 of the year of the ***Current Reporting Period***.
else

Locate the earliest ***Monitor Method*** for ALL locations in the monitor plan where ParameterCode = "NOXR", BeginDate is on or before the last day of the ***Current Reporting Period***, and EndDate is null or is on or after Jan 1 of the year of the ***Current Reporting Period***.

if found,
Set *EarliestMethodBeginDate* to the BeginDate for the located record in ***Monitor Method***.
else
Set *EarliestMethodBeginDate* to null.

If *EarliestMethodBeginDate* is NOT null,

Locate the earliest ***Location Program Record*** for location where ProgramCode is equal to "ARP", the Class is not equal to "NA", and UnitMonitorCertBeginDate is on or before the last day of the ***Current Reporting Period***, and the EndDate is null or is on or after Jan 1 of the year of the ***Current Reporting Period***.

If found,
If the EmissionsRecordingBeginDate is null,

If the later of *EarliestMethodBeginDate* and the UnitMonitorCertBeginDate is in a year prior to the current reporting period,
Set ***NOXR Start Quarter*** to 1.
else
Set ***NOXR Start Quarter*** to the quarter of the later of *EarliestMethodBeginDate* and UnitMonitorCertBeginDate .

Otherwise,
If the later of *EarliestMethodBeginDate* and the EmissionsRecordingBeginDate is in a year prior to the current reporting period,
Set ***NOXR Start Quarter*** to 1.
else
Set ***NOXR Start Quarter*** to the quarter of the later of *EarliestMethodBeginDate* and EmissionsRecordingBeginDate .

Locate the earliest ***Monitor Method*** for location where ParameterCode = "CO2" or "CO2M", BeginDate is on or before the last day of the ***Current Reporting Period***, and EndDate is null or is on or after Jan 1 of the year of the ***Current Reporting Period.***

If found,
If BeginDate is in a year prior to the current reporting period,
Set ***CO2 Start Quarter*** to 1.
else
Set ***CO2 Start Quarter*** to the quarter of the BeginDate.

If (the Quarter of the ***Current Reporting Period*** is greater than 2 OR (***Annual Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is equal to 2))

Locate the earliest ***Monitor Method*** for location where ParameterCode = "HI" or "HIT", MethodCode is not equal to "EXP", BeginDate is on or before the last day of the ***Current Reporting Period***, and EndDate is null or is on or after Jan 1 of the year of the ***Current Reporting Period***.

If found,
If BeginDate is in a year prior to the ***Current Reporting Period***,
If ***Annual Reporting Requirement*** == true
Set ***Heat Input Start Quarter*** to 1
else
Set ***Heat Input Start Quarter*** to 2

else if BeginDate is in Quarter 1 of the year of the ***Current Reporting Period*** AND ***Annual Reporting Requirement*** == false,

Set ***Heat Input Start Quarter*** to to 2

else

Set ***Heat Input Start Quarter*** to the quarter of the BeginDate.

Locate the earliest ***Monitor Method*** for location where ParameterCode is equal to "NOX" or "NOXM", BeginDate is on or before the last day of the ***Current Reporting Period***, and EndDate is null or is on or after Jan 1 of the year of the ***Current Reporting Period***.

If found,

If BeginDate is in a year prior to the ***Current Reporting Period***,

If ***Annual Reporting Requirement*** == true

Set ***NOX Start Quarter*** to 1

else

Set ***NOX Start Quarter*** to 2

else if BeginDate is in Quarter 1 of the year of the ***Current Reporting Period*** AND ***Annual Reporting Requirement*** == false,

Set ***NOX Start Quarter*** to to 2

else

Set ***NOX Start Quarter*** to the quarter of the BeginDate.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-2

Check Name: Compare SO2 Mass Accumulator Values

Related Former Checks: HOURCV-22

Applicability: General Check

Description: This check compares the accumulator reported SO2 Mass with the calculated value and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

SO2 Mass Quarterly Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "SO2M" AND
UOM = "TON"

Current SO2 Summary Value Record = SummaryValue record at this location where
Parameter = "SO2M" AND
Reporting Period ID = Current Reporting Period

if (***Rpt Period SO2 Mass Calculated Accumulator Array*** for this location == -1 OR ***Expected Summary Value SO2 Array*** for this location == false)
Rpt Period SO2 Mass Calculated Value = null
else
Rpt Period SO2 Mass Calculated Value = (***Rpt Period SO2 Mass Calculated Accumulator Array*** for this location / 2000, and rounded to one decimal place).

if (***Rpt Period SO2 Mass Reported Accumulator Array*** for this location >= 0)

Rpt Period SO2 Mass Reported Accumulator Array for this location = (***Rpt Period SO2 Mass Reported Accumulator Array*** for this location/ 2000, and rounded to one decimal place).

if (***Current SO2 Summary Value Record*** is null OR ***Current SO2 Summary Value Record***.Current Reporting Period Total is null)
if (***Expected Summary Value SO2 Array*** for this location == true)
return result C
else
if (***Expected Summary Value SO2 Array*** for this location == false)
if (***Rpt Period Op Hours Accumulator Array*** for this Location is not equal to 0 OR ***Current SO2 Summary Value Record***.Current Reporting Period Total is not equal to 0)
return result D
else
SO2 Mass Quarterly Reported Value = ***Current SO2 Summary Value Record***.Current Reporting Period Total

if (SO2 Mass Quarterly Reported Value < 0)
return result F

else if (SO2 Mass Quarterly Reported Value is not rounded to one decimal place)
return result G

else if (***Rpt Period SO2 Mass Calculated Value*** is not null)

If (***Rpt Period SO2 Mass Calculated Value*** <> SO2 Mass Quarterly Reported Value)
if (ABS(***Rpt Period SO2 Mass Calculated Value*** - SO2 Mass Quarterly Reported Value) > SO2 Mass Quarterly Tolerance)
return Result A
else

append "SO2M" to ***Emissions Tolerance Deviators***.

else

return result E

// if no result

if (***Rpt Period SO2 Mass Reported Accumulator Array*** for this location >= 0 AND ABS(***Rpt Period SO2 Mass Reported Accumulator Array*** for this location - SO2 Mass Quarterly Reported Value) > SO2 Mass Quarterly Tolerance)

Reported Emissions Value = ***Rpt Period SO2 Mass Reported Accumulator Array*** for this location

return result B

Results:

Result	Response	Severity
A	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for SO2M for the reporting period is inconsistent with the recalculated value of [calcval].	Critical Error Level 1
B	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for SO2M is inconsistent with [sum], the sum of the hourly values reported in the DHV records for the reporting period.	Critical Error Level 1
C	The CurrentReportingPeriodTotal in the Summary Value record for SO2M is missing or the record is missing.	Critical Error Level 1
D	You reported a value as the CurrentReportingPeriodTotal in the Summary Value record for SO2M, but there were no Hourly Operating Data records or appropriate SO2 Methods defined in your monitoring plan.	Critical Error Level 1
E	The CurrentReportingPeriodTotal in the Summary Value record for [param] could not be recalculated because of errors listed above.	Critical Error Level 1
F	The CurrentReportingPeriodTotal reported in the Summary Value record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
G	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-3

Check Name: Compare CO2 Mass Accumulator Values

Related Former Checks: HOURCV-23

Applicability: General Check

Description: This check compares the accumulator reported CO2 Mass with the calculated value and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

If ***Current Reporting Period Year*** is greater than or equal to 2012
CO2 Mass Quarterly Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "CO2M" AND
UOM = "TON"
else
CO2 Mass Quarterly Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "CO2M-OLD" AND
UOM = "TON"

Current CO2 Summary Value Record = SummaryValue record at this location where
Parameter = "CO2M" AND
Reporting Period ID = Current Reporting Period

if (***Rpt Period CO2 Mass Calculated Accumulator Array*** for this location < 0 OR ***Expected Summary Value CO2 Array*** for this location == false)
Rpt Period CO2 Mass Calculated Value = null
else
Rpt Period CO2 Mass Calculated Value = ***Rpt Period CO2 Mass Calculated Accumulator Array*** for this location, rounded to one decimal place.

if (***Rpt Period CO2 Mass Reported Accumulator Array*** for this location >= 0)
Rpt Period CO2 Mass Reported Accumulator Array for this location = (***Rpt Period CO2 Mass Reported Accumulator Array*** for this location, rounded to one decimal place).

if (***Current CO2 Summary Value Record*** is null OR ***Current CO2 Summary Value Record***.Current Reporting Period Total is null)
if (***Expected Summary Value CO2 Array*** for this location == true)
return result C
else
if (***Expected Summary Value CO2 Array*** for this location == false)
if (***Rpt Period Op Hours Accumulator Array*** for this Location is not equal to 0 OR ***Current CO2 Summary Value Record***.Current Reporting Period Total is not equal to 0)
return result D
else
CO2 Mass Quarterly Reported Value = ***Current CO2 Summary Value Record***.Current Reporting Period Total

if (CO2 Mass Quarterly Reported Value < 0)
return result F

else if (CO2 Mass Quarterly Reported Value is not rounded to one decimal place)
return result G

else if (***Rpt Period CO2 Mass Calculated Value*** is not null)
if (***Rpt Period CO2 Mass Calculated Value*** <> CO2 Mass Quarterly Reported Value)
if (ABS(***Rpt Period CO2 Mass Calculated Value*** - CO2 Mass Quarterly Reported Value) > CO2 Mass Quarterly Tolerance)

return Result A
else
append "CO2M" to ***Emissions Tolerance Deviators***.
else
if (***Rpt Period CO2 Mass Calculated Accumulator Array*** for this location == -1)
return result E

// if no result
if (***Rpt Period CO2 Mass Reported Accumulator Array*** for this location >= 0 AND ABS(***Rpt Period CO2 Mass Reported Accumulator Array*** for this location - CO2 Mass Quarterly Reported Value) > CO2 Mass Quarterly Tolerance)
Reported Emissions Value = ***Rpt Period CO2 Mass Reported Accumulator Array*** for this location
return Result B

Results:

Result	Response	Severity
A	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for CO2M for the reporting period is inconsistent with the recalculated value of [calcval].	Critical Error Level 1
B	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for CO2M is inconsistent with [sum], the sum of the hourly values reported in the DHV records for the reporting period.	Critical Error Level 1
C	The CurrentReportingPeriodTotal in the Summary Value record for CO2M is missing or the record is missing.	Critical Error Level 1
D	You reported a value for the CurrentReportingPeriodTotal in the Summary Value record for [param], but there was no emissions data in your file or an appropriate CO2 Method defined in your monitoring plan.	Critical Error Level 1
E	The CurrentReportingPeriodTotal in the Summary Value record for [param] could not be recalculated because of errors listed above.	Critical Error Level 1
F	The CurrentReportingPeriodTotal reported in the Summary Value record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
G	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-4

Check Name: Compare HI Accumulator Values

Related Former Checks: HOURCV-24

Applicability: General Check

Description: This check compares the accumulator reported HI with the calculated value and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

HI Quarterly Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "HIT" AND
UOM = "MMBTU"

Current HI Summary Value Record = SummaryValue record at this location where
Parameter = "HIT" AND
Reporting Period ID = Current Reporting Period

if (***Rpt Period HI Calculated Accumulator Array*** for this location == -1 OR ***Expected Summary Value HI Array*** for this location == false)
Rpt Period HI Calculated Value = null
else
Rpt Period HI Calculated Value = ***Rpt Period HI Calculated Accumulator Array*** for this location, rounded to zero decimal places.

if (***Rpt Period HI Reported Accumulator Array*** for this location >= 0)
Rpt Period HI Reported Accumulator Array for this location = (***Rpt Period HI Reported Accumulator Array*** for this location, rounded to zero decimal places).

if (***Current HI Summary Value Record*** is null OR ***Current HI Summary Value Record***.Current Reporting Period Total is null)
if (***Expected Summary Value HI Array*** for this location == true)
return result C
else
if (***Expected Summary Value HI Array*** for this location == false)
if (***Rpt Period Op Hours Accumulator Array*** for this Location is not equal to 0 OR ***Current HI Summary Value Record***.Current Reporting Period Total is not equal to 0)
return result D
else
HI Quarterly Reported Value = ***Current HI Summary Value Record***.Current Reporting Period Total

if (HI Quarterly Reported Value < 0)
return result F

else if (HI Quarterly Reported Value is not rounded to zero decimal places)
return result G

else if (***Rpt Period HI Calculated Value*** is not null)
if (***Rpt Period HI Calculated Value*** <> HI Quarterly Reported Value)
if (ABS(***Rpt Period HI Calculated Value*** - HI Quarterly Reported Value) > HI Quarterly Tolerance)
return result A
else
append "HIT" to ***Emissions Tolerance Deviators***.
else
return result E

// if no result
if (***Rpt Period HI Reported Accumulator Array*** for this location >= 0 AND ABS(***Rpt Period HI Reported Accumulator Array*** for this location (rounded to zero decimal places) - HI Quarterly Reported Value) > HI Quarterly Tolerance)
Reported Emissions Value = ***Rpt Period HI Reported Accumulator Array*** for this location
return Result B

Results:

Result	Response	Severity
A	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for HIT for the reporting period is inconsistent with the recalculated value of [calcval].	Critical Error Level 1
B	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for HIT is inconsistent with [sum], the sum of the hourly values reported in the DHV records for the reporting period.	Critical Error Level 1
C	The CurrentReportingPeriodTotal in the Summary Value record for HIT is missing or the record is missing.	Critical Error Level 1
D	You reported a value as the CurrentReportingPeriodTotal in the Summary Value record for HIT, but there were no Hourly Operating Data records or appropriate HI Methods defined in your monitoring plan.	Critical Error Level 1
E	The CurrentReportingPeriodTotal in the Summary Value record for [param] could not be recalculated because of errors listed above.	Critical Error Level 1
F	The CurrentReportingPeriodTotal reported in the Summary Value record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
G	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-5

Check Name: Compare Op Hours Values

Related Former Checks: HOURCV-25

Applicability: General Check

Description: This check compares the accumulator reported Op Hours with the calculated value and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

Op Hours Quarterly Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "OPHOURS" AND
UOM = "HR"

Current Op Hours Summary Value Record = SummaryValue record at this location where
Parameter = "OPHOURS" AND
Reporting Period ID = Current Reporting Period

if (***Rpt Period Op Hours Accumulator Array*** for this location == -1 OR (***LME HI Method*** is not null and location is a common pipe))
Rpt Period Op Hours Calculated Value = null
Rpt Period Op Days Calculated Value = null
else
Rpt Period Op Hours Calculated Value = ***Rpt Period Op Hours Accumulator Array*** for this location
Rpt Period Op Days Calculated Value = ***Rpt Period Op Days Accumulator Array*** for this location

if (***Current Op Hours Summary Value Record*** is null OR ***Current Op Hours Summary Value Record***.Current Reporting Period Total is null)

If (***LME HI Method*** is null or location is <u>not</u> a common pipe)
return result B
else
Op Hours Quarterly Reported Value = ***Current Op Hours Summary Value Record***.Current Reporting Period Total

if (Op Hours Quarterly Reported Value < 0)
return result D

else if (Op Hours Quarterly Reported Value is not rounded to zero decimal places)
return result E

else if (***Rpt Period Op Hours Calculated Value*** is not null)
if (***Rpt Period Op Hours Calculated Value*** <> Op Hours Quarterly Reported Value)
if (ABS(***Rpt Period Op Hours Calculated Value*** - Op Hours Quarterly Reported Value) > Op Hours Quarterly Tolerance)
return Result A
else
append "OPHOURS" to ***Emissions Tolerance Deviators***.

else
return result C

Results:

Result	Response	Severity
A	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for OPHOURS is inconsistent with [calcval], the number of operating hours reported in the Hourly Operating Data records for the reporting period.	Critical Error Level 1
B	The CurrentReportingPeriodTotal in the Summary Value record for OPHOURS is missing or the record is missing.	Critical Error Level 1
C	The CurrentReportingPeriodTotal in the Summary Value record for [param] could not be recalculated because of errors listed above.	Critical Error Level 1
D	The CurrentReportingPeriodTotal reported in the Summary Value record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
E	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-6

Check Name: Compare Op Time Values

Related Former Checks: HOURCV-26

Applicability: General Check

Description: This check compares the accumulator reported Op Time with the calculated value and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

Op Time Quarterly Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "OPTIME" AND
UOM = "HR"

Current Op Time Summary Value Record = SummaryValue record at this location where
Parameter = "OPTIME" AND
Reporting Period ID = Current Reporting Period

if (***Rpt Period Op Time Accumulator Array*** for this location == -1 OR (***LME HI Method*** is not null and location is a common pipe))
Rpt Period Op Time Calculated Value = null
else
Rpt Period Op Time Calculated Value = ***Rpt Period Op Time Accumulator Array*** for this location

if (***Current Op Time Summary Value Record*** is null OR ***Current Op Time Summary Value Record***.Current Reporting Period Total is null)
If (***LME HI Method*** is null or location is not a common pipe)
if (***Legacy Data Evaluation*** == true)
return result B
else
return result E
else
Op Time Quarterly Reported Value = ***Current Op Time Summary Value Record***.Current Reporting Period Total
if (Op Time Quarterly Reported Value >= 0)
if (Op Time Quarterly Reported Value is not rounded to two decimal places)
return result F
else

if (***Rpt Period Op Time Calculated Value*** is not null)
if (***Rpt Period Op Time Calculated Value*** <> Op Time Quarterly Reported Value)
if (ABS(***Rpt Period Op Time Calculated Value*** - Op Time Quarterly Reported Value) > Op Time Quarterly Tolerance)
return A
else
append "OPTIME" to ***Emissions Tolerance Deviators***.
else
return result D
else
return result C

Results:

Result	Response	Severity
A	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for OPTIME is inconsistent with [calcval], the sum of the hourly values reported in the Hourly Operating Data records for the reporting period.	Critical Error Level 1
B	You did not report a Summary Value record for OPTIME for the reporting period. While this information was not required for legacy EDR data, it is required for ECMPS.	Informational Message
C	The CurrentReportingPeriodTotal reported in the Summary Value record for OPTIME is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
D	The CurrentReportingPeriodTotal in the Summary Value record for [param] could not be recalculated because of errors listed above.	Critical Error Level 1
E	The CurrentReportingPeriodTotal in the Summary Value record for OPTIME is missing or the record is missing.	Critical Error Level 1
F	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-7

Check Name: Compare NOx Rate Accumulator Values

Related Former Checks: HOURCV-27

Applicability: General Check

Description: This check compares the accumulator reported NOx Rate with the calculated value and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

If (***Expected Summary Value NOx Rate Array*** for this location == true)
if (***LME Annual*** == true)
if (***Rpt Period HI Calculated Value*** is not null and ***Rpt Period NOx Mass Calculated Value*** is not null)
If (***Rpt Period NOx Mass Calculated Accumulator Array*** for this location = 0)
Rpt Period NOx Rate Calculated Value = 0
else
Rpt Period NOx Rate Calculated Value = ***Rpt Period NOx Mass Calculated Accumulator Array*** for this location / ***Rpt Period HI Calculated Value***, and round the result to three decimal places
else
Rpt Period NOx Rate Calculated Value = null
else
if (***Rpt Period NOx Rate Hours Accumulator Array*** for this location > 0 AND ***Rpt Period NOx Rate Calculated Accumulator Array*** for this location >= 0)
Rpt Period NOx Rate Calculated Value = ***Rpt Period NOx Rate Calculated Accumulator Array*** for this location / ***Rpt Period NOx Rate Hours Accumulator Array*** for the location, and round the result to three decimal places
Rpt Period NOx Rate Sum = ***Rpt Period NOx Rate Calculated Accumulator Array*** for this location
Rpt Period NOx Rate Hours = ***Rpt Period NOx Rate Hours Accumulator Array*** for this location

else if (***Rpt Period NOx Rate Hours Accumulator Array*** for this location == 0 AND ***Rpt Period NOx Rate Calculated Accumulator Array*** for this location == 0)
Rpt Period NOx Rate Calculated Value = 0
Rpt Period NOx Rate Sum = 0
Rpt Period NOx Rate Hours = 0
else
Rpt Period NOx Rate Calculated Value = null
Rpt Period NOx Rate Sum = null
Rpt Period NOx Rate Hours = null

if (***Rpt Period NOx Rate Hours Accumulator Array*** for this location > 0 AND ***Rpt Period NOx Rate Reported Accumulator Array*** for this location >= 0)
Rpt Period NOx Rate Reported Accumulator Array for this location = ***Rpt Period NOx Rate Reported Accumulator Array*** for this location / ***Rpt Period NOx Rate Hours Accumulator Array*** for this location, and round the result to three decimal places
else
Rpt Period NOx Rate Reported Accumulator Array for this location = -1
else
Rpt Period NOx Rate Calculated Value = null

NOx Rate Quarterly Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "NOXR" AND
UOM = "LBMMBTU"

Current NOx Rate Summary Value Record = SummaryValue record at this location where
Parameter = "NOXR" AND
Reporting Period ID = Current Reporting Period

if (***Current NOx Rate Summary Value Record*** is null OR ***Current NOx Rate Summary Value Record***.Current Reporting Period Total is null)

if (***Expected Summary Value NOx Rate Array*** for this location == true)

return result C

else

if (***Expected Summary Value NOx Rate Array*** for this location == false)

if (***Rpt Period NOx Rate Hours Accumulator Array*** for this Location is not equal to 0 OR ***Current NOx Rate Summary Value Record***.Current Reporting Period Total is not null)

return result D

else

NOx Rate Quarterly Reported Value = ***Current NOx Rate Summary Value Record***.Current Reporting Period Total

If (NOx Rate Quarterly Reported Value < 0)

return result F

else if (***Rpt Period NOx Rate Calculated Value*** is not null)

if (ABS(***Rpt Period NOx Rate Calculated Value*** - NOx Rate Quarterly Reported Value) > NOx Rate Quarterly Tolerance)

return result A

else

return result E

//if no result

if (***LME Annual*** == false)

if (***Current Monitor Plan Location Record***.LocationName begins with "MS" OR ***Multiple Stack Configuration*** == false)

if (***Rpt Period NOx Rate Reported Accumulator Array*** for this location >= 0 AND ABS(***Rpt Period NOx Rate Reported Accumulator Array*** for this location - NOx Rate Quarterly Reported Value) > NOx Rate Quarterly Tolerance)

Reported Emissions Value = ***Rpt Period NOx Rate Reported Accumulator Array*** for this location

return Result B

Results:

Result	Response	Severity
A	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for NOXR for the reporting period is inconsistent with the recalculated value of [calcval].	Critical Error Level 1
B	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for NOXR is inconsistent with [average], the average of the hourly values reported in the DHV records for the reporting period.	Critical Error Level 1
C	The CurrentReportingPeriodTotal in the Summary Value record for NOXR is missing or the record is missing.	Critical Error Level 1
D	You reported a value as the CurrentReportingPeriodTotal in the Summary Value record for NOXR, but this is not appropriate, either because there were no Hourly Operating Data records in your emissions file, or because this value is not consistent with the unit program records and monitoring methodologies in your monitoring plan.	Critical Error Level 1
E	The CurrentReportingPeriodTotal in the Summary Value record for [param] could not be recalculated because of errors listed above.	Critical Error Level 1
F	The CurrentReportingPeriodTotal reported in the Summary Value record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-8

Check Name: Compare NOx Mass Accumulator Values

Related Former Checks:

Applicability: General Check

Description: This check compares the accumulator reported NOx Mass with the calculated value and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

NOx Mass Quarterly Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "NOXM" AND
UOM = "TON"

Current NOx Mass Summary Value Record = SummaryValue record at this location where
Parameter = "NOXM" AND
Reporting Period ID = Current Reporting Period

if (***Rpt Period NOx Mass Calculated Accumulator Array*** for this location == -1 OR ***Expected Summary Value NOx Mass Array*** for this location == false)
Rpt Period NOx Mass Calculated Value = null
else
Rpt Period NOx Mass Calculated Value = ***Rpt Period NOx Mass Calculated Accumulator Array*** for this location / 2000, and rounded to one decimal place).

if (***Rpt Period NOx Mass Reported Accumulator Array*** for this location >= 0)
Rpt Period NOx Mass Reported Accumulator Array for this location = ***Rpt Period NOx Mass Reported Accumulator Array*** for this location/ 2000, and rounded to one decimal place).

if (***Current NOx Mass Summary Value Record*** is null OR ***Current NOx Mass Summary Value Record***.Current Reporting Period Total is null)
if (***Expected Summary Value NOxMass Array*** for this location == true)
return result C
else
if (***Expected Summary Value NOX Array*** for this location == false)
if (***Rpt Period Op Hours Accumulator Array*** for this Location is not equal to 0 OR ***Current NOx Mass Summary Value Record***.Current Reporting Period Total is not equal to 0)
return result D
else
NOx Mass Quarterly Reported Value = ***Current NOx Mass Summary Value Record***.Current Reporting Period Total

If (NOx Mass Quarterly Reported Value < 0)
return result F

else if (NOx Mass Quarterly Reported Value is not rounded to one decimal place)
return result G

else if (***Rpt Period NOx Mass Calculated Value*** is not null)
if (***Rpt Period NOx Mass Calculated Value*** <> NOx Mass Quarterly Reported Value)
if (ABS(***Rpt Period NOx Mass Calculated Value*** - NOx Mass Quarterly Reported Value) > NOx Mass Quarterly Tolerance)
return Result A
else
append "NOXM" to ***Emissions Tolerance Deviators***.

else
return result E

// if no result
If (***Rpt Period NOx Mass Reported Accumulator Array*** for this location >= 0 AND ABS(***Rpt Period NOx Mass Reported Accumulator Array*** for this location - NOx Mass Quarterly Reported Value) > NOx Mass Quarterly Tolerance)
Reported Emissions Value = ***Rpt Period NOx Mass Reported Accumulator Array*** for this location
return Result B

Results:

Result	Response	Severity
A	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for NOXM for the reporting period is inconsistent with the recalculated value of [calcval].	Critical Error Level 1
B	The CurrentReportingPeriodTotal of [sumval] reported in the Summary Value record for NOXM is inconsistent with [sum], the sum of the hourly values reported in the DHV records for the reporting period.	Critical Error Level 1
C	The CurrentReportingPeriodTotal in the Summary Value record for NOXM is missing or the record is missing.	Critical Error Level 1
D	You reported a value as the CurrentReportingPeriodTotal in the Summary Value record for NOXM, but there were no Hourly Operating Data records or appropriate NOX Methods defined in your monitoring plan.	Critical Error Level 1
E	The CurrentReportingPeriodTotal in the Summary Value record for [param] could not be recalculated because of errors listed above.	Critical Error Level 1
F	The CurrentReportingPeriodTotal reported in the Summary Value record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
G	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-10

Check Name: Compare CO2 Mass YTD Values

Related Former Checks:

Applicability: General Check

Description: This check compares the reported annual CO2 Mass with the calculated values and generates an error message if the difference is greater than the accepted tolerance.

Specifications:

Annual CO2M Calculated Value = null

if (***Rpt Period CO2 Mass Calculated Value*** is not null OR ***Expected Summary Value CO2 Array*** for this location == false)

if (***Expected Summary Value CO2 Array*** for this location == true)

If (***Emissions Tolerance Deviators*** contains "CO2M")
Annual CO2M Calculated Value = ***Current CO2 Summary Value Record***.Current Reporting Period Total
else
Annual CO2M Calculated Value = ***Rpt Period CO2 Mass Calculated Value***

else if (Quarter of the ***Current Reporting Period*** is greater than 1)
Annual CO2M Calculated Value = 0

If (Quarter of the ***Current Reporting Period*** is greater than 1)

If (***CO2 Start Quarter*** is not null)

For each quarter in the current year from the ***CO2 Start Quarter*** to the quarter prior to the quarter of the ***Current Reporting Period***:

Locate an Op Supp Data record for the location and quarter where ParameterCode = "CO2M".

If not found,
if (***Expected Summary Value CO2 Array*** for this location == true)
set ***Annual CO2M Calculated Value*** to null
return result A

Otherwise,
add OpValue to ***Annual CO2M Calculated Value.***

else
set ***Annual CO2M Calculated Value*** to null

if (***Current CO2 Summary Value Record*** is not null)
If (***Annual CO2M Calculated Value*** is null AND ***Expected Summary Value CO2 Array*** for this location == false)
return result G

else if (***Current CO2 Summary Value Record.***YearToDateTotal is null or is less than 0)
return result B

else if (***Current CO2 Summary Value Record.***YearToDateTotal is not rounded to one decimal place)
return result D

else if (***Annual CO2M Calculated Value*** is not null)

if (***Annual CO2M Calculated Value*** <> ***Current CO2 Summary Value Record.***YearToDateTotal)
return result C

// If no result
If (***Current CO2 Summary Value Record.***OzoneSeasonToDateTotal is not null)
return result E

else
If (***Expected Summary Value CO2 Array*** for this location == false AND ***Annual CO2M Calculated Value*** > 0)
return result F

Results:

Result	Response	Severity
A	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The [fieldname] in the Summary Value record for [param] is missing or invalid.	Critical Error Level 1
C	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc].	Critical Error Level 1
D	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
E	You reported OzoneSeasonToDate in the Summary Value record for [param], but this is not valid for this parameter.	Critical Error Level 1
F	You did not report a Summary Value record to report year-to-date total for [param].	Critical Error Level 1
G	You reported a Summary Value record for [param], but there was no [param] method defined in your monitoring plan that was active during the year.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-11

Check Name: Compare SO2 Mass YTD Values

Related Former Checks:

Applicability: General Check

Description: This check compares the reported annual SO2 Mass with the calculated values and generates an error message if the difference is greater than the accepted tolerance.

Specifications:

Annual SO2M Calculated Value = null

if (***Rpt Period SO2 Mass Calculated Value*** is not null OR ***Expected Summary Value SO2 Array*** for this location == false)

if (***Expected Summary Value SO2 Array*** for this location == true)
If (***Emissions Tolerance Deviators*** contains "SO2M")
Annual SO2M Calculated Value = ***Current SO2 Summary Value Record***.Current Reporting Period Total
else
Annual SO2M Calculated Value = ***Rpt Period SO2 Mass Calculated Value***

else if (Quarter of the ***Current Reporting Period*** is greater than 1)
Annual SO2M Calculated Value = 0

If (Quarter of the ***Current Reporting Period*** is greater than 1)

If (***SO2 Start Quarter*** is not null)

For each quarter from the ***SO2 Start Quarter*** to the quarter prior to the quarter of the ***Current Reporting Period***:

Locate an Op Supp Data record for the location and quarter where ParameterCode = "SO2M".

If not found,
if (***Expected Summary Value SO2 Array*** for this location == true)
set ***Annual SO2M Calculated Value*** to null
return result A

Otherwise,
add OpValue to ***Annual SO2M Calculated Value.***

else
set ***Annual SO2M Calculated Value*** to null

if (***Current SO2 Summary Value Record*** is not null)

if (***Annual SO2M Calculated Value*** is null AND ***Expected Summary Value SO2 Array*** for this location == false)
return result H

else if (***Current SO2 Summary Value Record.***YearToDateTotal is null or is less than 0)
return result B

else if (***Current SO2 Summary Value Record.***YearToDateTotal is not rounded to one decimal place)
return result D

else if (***Annual SO2M Calculated Value*** is not null)

if (***Annual SO2M Calculated Value*** <> ***Current SO2 Summary Value Record.***YearToDateTotal)
return result C

// if no result

if (***Current SO2 Summary Value Record.***OzoneSeasonToDateTotal is not null)
return result F

else if (***LME Annual*** is equal to true and ***Current SO2 Summary Value Record.***YearToDateTotal is greater than 25)
return result E

else

If (***Expected Summary Value SO2 Array*** for this location == false AND ***Annual SO2M Calculated Value*** > 0)
return result G

Results:

Result	Response	Severity
A	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The [fieldname] in the Summary Value record for [param] is missing or invalid.	Critical Error Level 1
C	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc].	Critical Error Level 1
D	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
E	The [paramname] emissions from this unit exceed the applicable number of tons necessary to qualify as an LME unit. According to Part 75.19(b), you must install the appropriate monitoring systems to measure [paramname] by December 31 of the year following this reporting period.	Informational Message
F	You reported OzoneSeasonToDate in the Summary Value record for [param], but this is not valid for this parameter.	Critical Error Level 1
G	You did not report a Summary Value record to report year-to-date total for [param].	Critical Error Level 1
H	You reported a Summary Value record for [param], but there was no [param] method defined in your monitoring plan that was active during the year.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-12

Check Name: Compare NOx Mass YTD and OS Values

Related Former Checks:

Applicability: General Check

Description: This check compares the reported annual and ozone-season NOx Mass with the calculated values and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

Annual NOXM Calculated Value = null
OS NOXM Calculated Value = null
NOXM Summary Invalid Fields = null
Imprecise Fields = null

if (***Rpt Period NOx Mass Calculated Value*** is not null OR ***Expected Summary Value NOx Mass Array*** for this location == false)

if (***Expected Summary Value NOx Mass Array*** for this location == true)

If (***Annual Reporting Requirement*** == true)
If (***Emissions Tolerance Deviators*** contains "NOXM")
Annual NOXM Calculated Value = ***Current NOX Mass Summary Value Record***.Current Reporting Period Total
else
Annual NOXM Calculated Value = ***Rpt Period NOx Mass Calculated Value***

If (***OS Reporting Requirement*** == true)
if (Quarter of the ***Current Reporting Period*** is equal to 2 or 3)
If (***Annual Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is equal to 2)
OS NOXM Calculated Value = (***Rpt Period NOx Mass Calculated Accumulator Array*** for this location - ***April NOx Mass Calculated Accumulator Array*** for this location) / 2000, rounded to one decimal place.
else
If (***Emissions Tolerance Deviators*** contains "NOXM")
OS NOXM Calculated Value = ***Current NOX Mass Summary Value Record***.Current Reporting Period Total
else
OS NOXM Calculated Value = ***Rpt Period NOx Mass Calculated Value***

else if (Quarter of the ***Current Reporting Period*** is equal to 4)
OS NOXM Calculated Value = 0

else

If (***Annual Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is greater than 1)
Annual NOXM Calculated Value = 0

If (***OS Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is greater than 2)
OS NOXM Calculated Value = 0

If (the Quarter of the ***Current Reporting Period*** is greater than 2 OR (***Annual Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is equal to 2))

If (***NOX Start Quarter*** is not null)

For each quarter in the current year from the ***NOX Start Quarter*** to the quarter prior to the quarter of the ***Current Reporting Period***:

If this quarter is equal to 2 AND ***OS Reporting Requirement*** == true,
Locate an Op Supp Data record for the location and quarter where ParameterCode = "NOXMOS".

If not found,
if (***Expected Summary Value NOx Mass Array*** for this location == true)
set ***Annual NOXM Calculated Value*** to null.
set ***OS NOXM Calculated Value*** to null
return result A

otherwise,
Locate an Op Supp Data record for the location and quarter where ParameterCode = "NOXM".

If found,
set ***Annual NOXM Calculated Value*** to null.
set ***OS NOXM Calculated Value*** to null
return result A
Otherwise,
add OpValue to ***OS NOXM Calculated Value.***

If this quarter is not equal to 2 OR ***Annual Reporting Requirement*** == true,
Locate an Op Supp Data record for the location and quarter where ParameterCode = "NOXM".

If not found,
if (***Expected Summary Value NOx Mass Array*** for this location == true)
set ***Annual NOXM Calculated Value*** to null.
set ***OS NOXM Calculated Value*** to null
return result B

Otherwise,
if ***Annual Reporting Requirement*** == true
add OpValue to ***Annual NOXM Calculated Value.***

if this quarter is equal to 3 AND ***OS Reporting Requirement*** == true
add OpValue to ***OS NOXM Calculated Value.***

else
set ***Annual NOXM Calculated Value*** to null
set *OS NOXM Calculated Value* to null

if (***Current NOX Mass Summary Value Record*** is not null)

If (***OS NOXM Calculated Value Value*** is null AND ***Annual NOXM Calculated Value*** is null AND ***Expected Summary Value NOx Mass Array*** for this location == false)
return result K

Otherwise,
If (***Current NOX Mass Summary Value Record.***YearToDateTotal is null and ***Annual Reporting Requirement*** == true) OR (***Current NOX Mass Summary Value Record.***YearToDateTotal is less than 0,
append "YearToDateTotal" to ***NOXM Summary Invalid Fields***

If (***Current NOX Mass Summary Value Record.***OzoneSeasonToDateTotal is null and ***OS Reporting***

Requirement == true AND Quarter of the ***Current Reporting Period*** is equal to 2 or 3 or 4), OR ***Current NOX Mass Summary Value Record.***OzoneSeasonToDateTotal is less than 0,
append "OzoneSeasonToDateTotal" to ***NOXM Summary Invalid Fields***

If (***Current NOX Mass Summary Value Record.***YearToDateTotal is not rounded to one decimal place)
append "YearToDateTotal" to *Imprecise Fields*

If (***Current NOX Mass Summary Value Record.***OzoneSeasonToDateTotal is not rounded to one decimal place)
append "OzoneSeasonToDateTotal" to *Imprecise Fields*

If (***NOXM Summary Invalid Fields*** is not null)
return result C

else if (*Imprecise Fields* is not null)
Set ***NOXM Summary Invalid Fields*** to *Imprecise Fields*
return result E

else if (***Annual NOXM Calculated Value*** is not null OR ***OS NOXM Calculated Value*** is not null)

Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "NOXM" AND
UOM = "TON"

if (***Annual NOXM Calculated Value*** is not null AND ***Annual NOXM Calculated Value*** <> ***Current NOX Mass Summary Value Record.***YearToDateTotal)
append "YearToDateTotal" to ***NOXM Summary Invalid Fields***

if (***OS NOXM Calculated Value*** is not null AND ***OS NOXM Calculated Value*** <> ***Current NOX Mass Summary Value Record.***OzoneSeasonToDateTotal)

if (ABS(***OS NOXM Calculated Value - Current NOXM Summary Value Record.***OzoneSeasonToDateTotal) > Tolerance OR quarter of the ***Current Reporting Period*** is greater than 2)
append "OzoneSeasonToDateTotal" to ***NOXM Summary Invalid Fields***

If ***NOXM Summary Invalid Fields*** is not null,
If (***NOXM Summary Invalid Fields*** contains "Year")
If (***NOXM Summary Invalid Fields*** contains "Ozone")
return result D
else
return result H
else
return result I

// if no result

if (***OS Reporting Requirement*** == false and ***Current NOXM Summary Value Record.***OzoneSeasonToDateTotal is not null)
return result G

else if (***Annual Reporting Requirement*** == false and ***Current NOXM Summary Value Record***.YearToDateTotal is not null)
return result L

else if ((***LME Annual*** is equal to true and ***Current NOXM Summary Value Record.***YearToDateTotal is greater than 100) OR (***LME OS*** is equal to true and ***Current NOXM Summary Value Record.***OzoneSeasonToDateTotal is greater than 50))

return result F

else

If (***Expected Summary Value NOx Mass Array*** for this location == false AND (***OS NOXM Calculated Value*** > 0 OR ***Annual NOXM Calculated Value*** > 0))

return result J

Results:

Result	Response	Severity
A	The program could not determine ozone-season-to-date totals for [osparam], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
C	The [fieldname] in the Summary Value record for [param] is missing or invalid.	Critical Error Level 1
D	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc], and the OzoneSeasonToDateTotal of [osval] in the Summary Value record for [param] is inconsistent with the recalculated value of [oscalc].	Critical Error Level 1
E	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
F	The [paramname] emissions from this unit exceed the applicable number of tons necessary to qualify as an LME unit. According to Part 75.19(b), you must install the appropriate monitoring systems to measure [paramname] by December 31 of the year following this reporting period.	Informational Message
G	You reported OzoneSeasonToDate in the Summary Value record for [param], but this is not valid for locations that are not associated with an ozone-season program.	Critical Error Level 1
H	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc].	Critical Error Level 1
I	The OzoneSeasonToDateTotal of [osval] in the Summary Value record for [param] is inconsistent with the recalculated value of [oscalc].	Critical Error Level 1
J	You did not report a Summary Value record to report year-to-date total for [param].	Critical Error Level 1
K	You reported a Summary Value record for [param], but there was no [param] method defined in your monitoring plan that was active during the year.	Critical Error Level 1
L	You reported YearToDate in the Summary Value record for [param], but this is not valid for locations that only report during the ozone season.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-13

Check Name: Compare NOx Rate YTD Values

Related Former Checks:

Applicability: General Check

Description: This check compares the reported annual and ozone-season NOx Rate with the calculated values and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

Annual NOXR Calculated Value = null

If (***LME Annual*** == true)

Set *Total NOx Mass* to null.

if (***Expected Summary Value NOx Rate Array*** for this location == true)

if (***Rpt Period NOx Mass Calculated Accumulator Array*** for this location is greater than or equal to 0 AND ***Rpt Period HI Calculated Value*** is not null)

Set *Total NOx Mass* to ***Rpt Period NOx Mass Calculated Accumulator Array*** for this location.
Set *Total HI* to ***Rpt Period HI Calculated Value***.

else if (Quarter of the ***Current Reporting Period*** is greater than 1)
Set *Total NOx Mass* to 0.
Set *Total HI* to 0.

If (Quarter of the ***Current Reporting Period*** is greater than 1 AND *Total NOx Mass* is not null)

if (***NOXR Start Quarter*** is not null)

For each quarter in the current year from the ***NOXR Start Quarter*** to the quarter prior to the quarter of the ***Current Reporting Period***:

Locate an Op Supp Data record for the location and quarter where ParameterCode = "NOXR".

If not found,
if (***Expected Summary Value NOx Rate Array*** for this location == true)
set *Total NOx Mass* to null.
return result A

Otherwise,
set *NOX Value* to OpValue.

Locate an Op Supp Data record for the location and quarter where ParameterCode = "HIT".

If not found,
if (***Expected Summary Value NOx Rate Array*** for this location == true)
set *Total HI* to null.
return result E

Otherwise,
Add OpValue to *Total HI*.
Calculate *NOX Value* = *NOX Value* * OpValue, and round the result to 1 decimal place.

Add *NOX Value* to *Total NOx Mass*.

else
Set *Total NOx Mass* to null.

If (*Total NOx Mass* is not null AND *Total HI* is not null)
If (*Total NOx Mass* == 0)
Set ***Annual NOXR Calculated Value*** to 0.
else
Calculate ***Annual NOXR Calculated Value*** = *Total NOx Mass* / *TotalHI*, and round the result to 3 decimal places.

else

Set *TotalOpHours* to null.

if (***Expected Summary Value NOx Rate Array*** for this location == true)

if (***Rpt Period NOx Rate Calculated Value*** is not null)

Annual NOXR Calculated Value = ***Rpt Period NOx Rate Sum***
Set *TotalOpHours* to ***Rpt Period NOx Rate Hours***.

else if ((Quarter of the ***Current Reporting Period*** is greater than 1)
Set *TotalOpHours* to 0.

If (Quarter of the ***Current Reporting Period*** is greater than 1 AND *TotalOpHours* is not null)

if (***NOXR Start Quarter*** is not null)

For each quarter from the ***NOXR Start Quarter*** to the quarter prior to the quarter of the ***Current Reporting Period***:

Locate an Op Supp Data record for the location and quarter where ParameterCode = "NOXRSUM".

If found,
Add OpValue to ***Annual NOXR Calculated Value.***

Locate an Op Supp Data record for the location and quarter where ParameterCode = "NOXRHRS".

If found,
Add OpValue to *TotalOpHours*.

If not found,
set ***Annual NOXR Calculated Value*** to null
return result A

Otherwise,

Locate an Op Supp Data record for the location and quarter where ParameterCode = "NOXR".

If not found,
if (***Expected Summary Value NOx Rate Array*** for this location == true)
set ***Annual NOXR Calculated Value*** to null
return result A

Otherwise,

set *NOXVal* to OpValue

Locate an Op Supp Data record for the location and quarter where ParameterCode = "OPHOURS" and FuelCode is null.

If not found,
if (***Expected Summary Value NOx Rate Array*** for this location == true)
set ***Annual NOXR Calculated Value*** to null
return result B

Otherwise,
Add OpValue to *TotalOpHours.*
Add OpValue * *NOXVal to* ***Annual NOXR Calculated Value***

else
set ***Annual NOXR Calculated Value*** to null.

If (***Annual NOXR Calculated Value*** is not null)
If (*TotalOpHours* == 0)
Set ***Annual NOXR Calculated Value*** to 0.

else if (***Annual NOXR Calculated Value*** > 0)
Calculate ***Annual NOXR Calculated Value*** = ***Annual NOXR Calculated Value*** / *TotalOpHours*, and round the result to 3 decimal places.

if (***Current NOXR Summary Value Record*** is not null)
If (***Annual NOXR Calculated Value*** is null AND ***Expected Summary Value NOx Rate Array*** for this location == false)
return result H

else if (***Current NOXR Summary Value Record.***YearToDateTotal is null or is less than 0)
return result C

else if (***Annual NOXR Calculated Value*** is not null)

if (***Annual NOXR Calculated Value*** <> ***Current NOXR Summary Value Record.***YearToDateTotal)

Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "NOXR" AND
UOM = "LBMMBTU"

if (ABS(***Annual NOXR Calculated Value - Current NOXR Summary Value Record.***YearToDateTotal) > Tolerance)
return result D

// if no result

if (***Current NOXR Summary Value Record.***OzoneSeasonToDateTotal is not null)
return result F

else
If (***Expected Summary Value NOx Rate Array*** for this location == false AND ***Annual NOXR Calculated Value*** > 0)
return result G

Results:

Result	Response	Severity
A	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The program could not determine year-to-date for [param], because the Op Supp Data record for OPHOURS is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
C	The [fieldname] in the Summary Value record for [param] is missing or invalid.	Critical Error Level 1
D	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc].	Critical Error Level 1
E	The program could not determine year-to-date for [param], because the Op Supp Data record for HIT is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
F	You reported OzoneSeasonToDate in the Summary Value record for [param], but this is not valid for this parameter.	Critical Error Level 1
G	You did not report a Summary Value record to report year-to-date total for [param].	Critical Error Level 1
H	You reported a Summary Value record for NOXR, but this is not appropriate, because this record is not consistent with the unit program records and monitoring methodologies in your monitoring plan. You only report a NOXR Summary Value if the unit belongs to the Acid Rain program.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-14

Check Name: Compare Total Heat Input YTD and OS Values

Related Former Checks:

Applicability: General Check

Description: This check compares the reported annual and ozone-season heat input with the calculated values and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

Annual HIT Calculated Value = null
OS HIT Calculated Value = null
HI Summary Invalid Fields = null
Imprecise Fields = null

if (***Rpt Period HI Calculated Value*** is not null OR ***Expected Summary Value HI Array*** for this location == false)

if (***Expected Summary Value HI Array*** for this location == true)

If (***Annual Reporting Requirement*** == true)
If (***Emissions Tolerance Deviators*** contains "HIT")
Annual HI Calculated Value = ***Current HI Summary Value Record***.Current Reporting Period Total
else
Annual HI Calculated Value = ***Rpt Period HI Calculated Value***

If (***OS Reporting Requirement*** == true)
if (the Quarter of the ***Current Reporting Period*** is equal to 2 or 3)
If (***Annual Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is equal to 2)
OS HIT Calculated Value = ***Rpt Period HI Calculated Accumulator Array*** for this location - ***April HI Calculated Accumulator Array*** for this location, and round the result to zero decimal places.
else
If (***Emissions Tolerance Deviators*** contains "HIT")
OS HIT Calculated Value = ***Current HI Summary Value Record***.Current Reporting Period Total
else
OS HIT Calculated Value = ***Rpt Period HI Calculated Value***

else if (Quarter of the ***Current Reporting Period*** is equal to 4)
OS HIT Calculated Value = 0

else

If (***Annual Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is greater than 1)
Annual HI Calculated Value = 0

If (***OS Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is greater than 2)
OS HIT Calculated Value = 0

If (the Quarter of the ***Current Reporting Period*** is greater than 2 OR (***Annual Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is equal to 2))

If (***Heat Input Start Quarter*** is not null)

For each quarter in the current year from the ***Heat Input Start Quarter*** to the quarter prior to the quarter of the

Current Reporting Period:

If this quarter is equal to 2 AND ***OS Reporting Requirement*** == true,
Locate an Op Supp Data record for the location and quarter where ParameterCode = "HITOS".

If not found,
if (***Expected Summary Value HI Array*** for this location == true)
set ***Annual HIT Calculated Value*** to null.
set ***OS HIT Calculated Value*** to null
return result A

otherwise,
Locate an Op Supp Data record for the location and quarter where ParameterCode = "HIT".

If found,
set ***Annual HIT Calculated Value*** to null.
set ***OS HIT Calculated Value*** to null
return result A

Otherwise,
add OpValue to ***OS HIT Calculated Value.***

If this quarter is not equal to 2 OR ***Annual Reporting Requirement*** == true,
Locate an Op Supp Data record for the location and quarter where ParameterCode = "HIT".

If not found,
if (***Expected Summary Value HI Array*** for this location == true)
set ***Annual HIT Calculated Value*** to null.
set ***OS HIT Calculated Value*** to null
return result B

Otherwise,
if ***Annual Reporting Requirement*** == true
add OpValue to ***Annual HIT Calculated Value.***

if this quarter is equal to 3 AND ***OS Reporting Requirement*** == true
add OpValue to ***OS HIT Calculated Value.***

else
set ***Annual HIT Calculated Value*** to null
set ***Annual OS HIT Calculated Value*** to null

if (***Current HI Summary Value Record*** is not null)

If (***OS HIT Calculated Value*** is null AND ***Annual HIT Calculated Value*** is null AND ***Expected Summary Value HI Array*** for this location == false and (***LME HI Method*** <> "LTFF" or location does not start with "CP"))
return result K

Otherwise,

If (***Current HI Summary Value Record.***YearToDateTotal is null and ***Annual Reporting Requirement*** == true)
OR (***Current HI Summary Value Record.***YearToDateTotal is less than 0,
append "YearToDateTotal" to ***HIT Summary Invalid Fields***

If (***Current HI Summary Value Record.***OzoneSeasonToDateTotal is null and ***OS Reporting Requirement*** == true
AND Quarter of the ***Current Reporting Period*** is equal to 2 or 3 or 4), OR ***Current HI Summary Value***

Record.OzoneSeasonToDateTotal is less than 0,
append "OzoneSeasonToDateTotal" to ***HIT Summary Invalid Fields***

If (***Current HI Summary Value Record.***YearToDateTotal is not rounded to zero decimal places)
append "YearToDateTotal" to *Imprecise Fields*

If (***Current HI Summary Value Record.***OzoneSeasonToDateTotal is not rounded to zero decimal places)
If (***Legacy Data Evaluation*** == false OR ***Current HI Summary Value Record.***OzoneSeasonToDateTotal is not rounded to one decimal place)
append "OzoneSeasonToDateTotal" to *Imprecise Fields*

If (***HIT Summary Invalid Fields*** is not null)
return result C

else if (*Imprecise Fields* is not null)
set ***HIT Summary Invalid Fields*** to *Imprecise Fields*
return result E

else if (***Annual HIT Calculated Value*** is not null OR ***OS HIT Calculated Value*** is not null)

Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "HIT" AND
UOM = "MMBTU"

if (***Annual HIT Calculated Value*** is not null AND ***Annual HIT Calculated Value*** <> ***Current HI Summary Value Record.***YearToDateTotal)
append "YearToDateTotal" to ***HIT Summary Invalid Fields***

if (***OS HIT Calculated Value*** is not null AND ***OS HIT Calculated Value*** <> ***Current HI Summary Value Record.***OzoneSeasonToDateTotal)

If (***Legacy Data Evaluation*** == false)

if (ABS(***OS HIT Calculated Value - Current HI Summary Value Record.***OzoneSeasonToDateTotal) > Tolerance OR the quarter of the ***Current Reporting Period*** is greater than 2)
append "OzoneSeasonToDateTotal" to ***HIT Summary Invalid Fields***

else
if (ABS(***OS HIT Calculated Value - Current HI Summary Value Record.***OzoneSeasonToDateTotal rounded to the nearest integer) > Tolerance OR the quarter of the ***Current Reporting Period*** is greater than 2)
append "OzoneSeasonToDateTotal" to ***HIT Summary Invalid Fields***

If (***HIT Summary Invalid Fields*** is not null)
If (***HIT Summary Invalid Fields*** contains "Year")
If (***HIT Summary Invalid Fields*** contains "Ozone")
return result D
else
return result H
else
If (***Legacy Data Evaluation*** == true)
return result F
else
return result I

// if no result

if (***OS Reporting Requirement*** == false and ***Current HI Summary Value Record.***OzoneSeasonToDateTotal is not null)
return result G

else if (***Annual Reporting Requirement*** == false and ***Current HI Summary Value Record.***YearToDateTotal is not null)
return result L

else
If (***Expected Summary Value HI Array*** for this location == false AND (***Annual HIT Calculated Value*** > 0 OR ***OS HIT Calculated Value*** > 0))
return result J

Results:

Result	Response	Severity
A	The program could not determine ozone-season-to-date totals for [osparam], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
C	The [fieldname] in the Summary Value record for [param] is missing or invalid.	Critical Error Level 1
D	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc], and the OzoneSeasonToDateTotal of [osval] in the Summary Value record for [param] is inconsistent with the recalculated value of [oscalc].	Critical Error Level 1
E	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
F	The OzoneSeasonToDateTotal of [osval] in the Summary Value record for [param] is inconsistent with the recalculated value of [oscalc].	Informational Message
G	You reported OzoneSeasonToDate in the Summary Value record for [param], but this is not valid for locations that are not associated with an ozone-season program.	Critical Error Level 1
H	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc].	Critical Error Level 1
I	The OzoneSeasonToDateTotal of [osval] in the Summary Value record for [param] is inconsistent with the recalculated value of [oscalc].	Critical Error Level 1
J	You did not report a Summary Value record to report year-to-date total for [param].	Critical Error Level 1
K	You reported a Summary Value record for [param], but there was no [param] method defined in your monitoring plan that was active during the year.	Critical Error Level 1
L	You reported YearToDate in the Summary Value record for [param], but this is not valid for locations that only report during the ozone season.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-15

Check Name: Compare Operating Time YTD and OS Values

Related Former Checks:

Applicability: General Check

Description: This check compares the reported annual and ozone-season operating time with the calculated values and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

Annual OPTIME Calculated Value = null
OS OPTIME Calculated Value = null
OPTIME Summary Invalid Fields = null
Imprecise Fields = null

if (***Rpt Period Op Time Calculated Value*** is not null)

If ***Annual Reporting Requirement*** == true
If (***Emissions Tolerance Deviators*** contains "OPTIME")
Annual OPTIME Calculated Value = ***Current Op Time Summary Value Record***.Current Reporting Period Total
else
Annual OPTIME Calculated Value = ***Rpt Period Op Time Calculated Value***

If OS Reporting Requirement == true)
if (the Quarter of the ***Current Reporting Period*** is equal to 2 or 3)
If ***Annual Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is equal to 2,
OS OPTIME Calculated Value = ***Rpt Period Op Time Accumulator Array*** for this location - ***April Op Time Accumulator Array*** for this location.
else
If (***Emissions Tolerance Deviators*** contains "OPTIME")
OS OPTIME Calculated Value = ***Current Op Time Summary Value Record***.Current Reporting Period Total
else
OS OPTIME Calculated Value = ***Rpt Period Op Time Calculated Value***

else if (Quarter of the ***Current Reporting Period*** is equal to 4)
OS OPTIME Calculated Value = 0

If (the Quarter of the ***Current Reporting Period*** is greater than 2 OR (***Annual Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is equal to 2))

If (***Start Quarter*** is not null)

For each quarter in the current year from the ***Start Quarter*** to the quarter prior to the quarter of the ***Current Reporting Period***:

If this quarter is equal to 2 AND ***OS Reporting Requirement*** == true
Locate an Op Supp Data record for the location and quarter where ParameterCode = "OSTIME".

If not found,
set ***Annual OPTIME Calculated Value*** to null.
set ***OS OPTIME Calculated Value*** to null

Locate the ***Facility*** record for the location.

If the First ECMPS Reporting Period in the retrieved record is not null AND is on or before the 2nd quarter of the current year,
return result A.

exit for.

Otherwise,
add OpValue to ***OS OPTIME Calculated Value.***

If this quarter is not equal to 2 OR ***Annual Reporting Requirement*** == true,
Locate an Op Supp Data record for the location and quarter where ParameterCode = "OPTIME".

If not found,
if ***Annual Reporting Requirement*** == true
set ***Annual OPTIME Calculated Value*** to null.

if ***OS Reporting Requirement*** == true
set ***OS OPTIME Calculated Value*** to null.

Locate the ***Facility*** record for the location.
If the First ECMPS Reporting Period in the retrieved record is not null AND is on or before the *Start Quarter* of the current year,
return result B.

exit for.

Otherwise,
if ***Annual Reporting Requirement*** == true
add OpValue to ***Annual OPTIME Calculated Value.***

if this quarter is equal to 3 AND ***OS Reporting Requirement*** == true
add OpValue to ***OS OPTIME Calculated Value.***

Otherwise,
set ***Annual OPTIME Calculated Value*** to null.
set ***OS OPTIME Calculated Value*** to null.

if (***Current Op Time Summary Value Record*** is not null)

If (***Current Op Time Summary Value Record.***YearToDateTotal is null and ***Annual Reporting Requirement*** == true) OR (***Current Op Time Summary Value Record.***YearToDateTotal is less than 0,
append "YearToDateTotal" to ***OPTIME Summary Invalid Fields***

If (***Current Op Time Summary Value Record.***OzoneSeasonToDateTotal is null and ***OS Reporting Requirement*** == true AND Quarter of the ***Current Reporting Period*** is equal to 2 or 3 or 4), OR ***Current Op Time Summary Value Record.***OzoneSeasonToDateTotal is less than 0,
append "OzoneSeasonToDateTotal" to ***OPTIME Summary Invalid Fields***

If (***Current Op Time Summary Value Record.***YearToDateTotal is not rounded to two decimal places)
append "YearToDateTotal" to *Imprecise Fields*

If (***Current Op Time Summary Value Record.***OzoneSeasonToDateTotal is not rounded to two decimal places)
append "OzoneSeasonToDateTotal" to *Imprecise Fields*

If (***OPTIME Summary Invalid Fields*** is not null)
return result C

else if (*Imprecise Fields* is not null)
Set ***OPTIME Summary Invalid Fields*** to *Imprecise Fields*
return result E

else if (***Annual OPTIME Calculated Value*** is not null OR ***OS OPTIME Calculated Value*** is not null)

Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "OPTIME" AND
UOM = "HR"

if (***Annual OPTIME Calculated Value*** is not null AND ***Annual OPTIME Calculated Value*** <> ***Current Op Time Summary Value Record.***YearToDateTotal)
append "YearToDateTotal" to ***OPTIME Summary Invalid Fields***

if (***OS OPTIME Calculated Value*** is not null AND ***OS OPTIME Calculated Value*** <> ***Current Op Time Summary Value Record.***OzoneSeasonToDateTotal)

if (ABS(***OS OPTIME Calculated Value - Current Op Time Summary Value Record.***OzoneSeasonToDateTotal) > Tolerance OR quarter of the ***Current Reporting Period*** is greater than 2)
append "OzoneSeasonToDateTotal" to ***OPTIME Summary Invalid Fields***

If ***OPTIME Summary Invalid Fields*** is not null,
If (***OPTIME Summary Invalid Fields*** contains "Year")
If (***OPTIME Summary Invalid Fields*** contains "Ozone")
return result D
else
return result G
else
return result H

// if no result

if (***OS Reporting Requirement*** == false and ***Current Op Time Summary Value Record.***OzoneSeasonToDateTotal is not null)
return result F

else if (***Annual Reporting Requirement*** == false and ***Current Op Time Summary Value Record.***YearToDateTotal is not null)
return result I

Results:

Result	Response	Severity
A	The program could not determine ozone-season-to-date totals for [osparam], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
C	The [fieldname] in the Summary Value record for [param] is missing or invalid.	Critical Error Level 1
D	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc], and the OzoneSeasonToDateTotal of [osval] in the Summary Value record for [param] is inconsistent with the recalculated value of [oscalc].	Critical Error Level 1
E	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
F	You reported OzoneSeasonToDate in the Summary Value record for [param], but this is not valid for locations that are not associated with an ozone-season program.	Critical Error Level 1
G	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc].	Critical Error Level 1
H	The OzoneSeasonToDateTotal of [osval] in the Summary Value record for [param] is inconsistent with the recalculated value of [oscalc].	Critical Error Level 1
I	You reported YearToDate in the Summary Value record for [param], but this is not valid for locations that only report during the ozone season.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-16

Check Name: Compare Operating Hours YTD and OS Values

Related Former Checks:

Applicability: General Check

Description: This check compares the reported annual and ozone-season operating hours with the calculated values and generates an error message if the difference is greater than the accepted tolerance.

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

Annual OPHOURS Calculated Value = null
OS OPHOURS Calculated Value = null
OPHOURS Summary Invalid Fields = null
Imprecise Fields = null

if (***Rpt Period Op Hours Calculated Value*** is not null)

If ***Annual Reporting Requirement*** == true
If (***Emissions Tolerance Deviators*** contains "OPHOURS")
Annual OPHOURS Calculated Value = ***Current Op Hours Summary Value Record***.Current Reporting Period Total
else
Annual OPHOURS Calculated Value = ***Rpt Period Op Hours Calculated Value***

If OS Reporting Requirement == true)
if (the Quarter of the ***Current Reporting Period*** is equal to 2 or 3)
If ***Annual Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is equal to 2,
OS OPHOURS Calculated Value = ***Rpt Period Op Hours Accumulator Array*** for this location - ***April Op Hours Accumulator Array*** for this location.
else
If (***Emissions Tolerance Deviators*** contains "OPHOURS")
OS OPHOURS Calculated Value = ***Current Op Hours Summary Value Record***.Current Reporting Period Total
else
OS OPHOURS Calculated Value = ***Rpt Period Op Hours Calculated Value***

else if (Quarter of the ***Current Reporting Period*** is equal to 4)
OS OPHOURS Calculated Value = 0

If (the Quarter of the ***Current Reporting Period*** is greater than 2 OR (***Annual Reporting Requirement*** == true AND the Quarter of the ***Current Reporting Period*** is equal to 2))

If (***Start Quarter*** is not null)

For each quarter in the current year from the *Start Quarter* to the quarter prior to the quarter of the ***Current Reporting Period***:

If this quarter is equal to 2 AND ***OS Reporting Requirement*** == true,
Locate an Op Supp Data record for the location and quarter where ParameterCode = "OSHOURS" and FuelCd is null.

If not found,
set ***Annual OPHOURS Calculated Value*** to null.
set ***OS OPHOURS Calculated Value*** to null
return result A

Otherwise,
add OpValue to ***OS OPHOURS Calculated Value.***

If this quarter is not equal to 2 OR ***Annual Reporting Requirement*** == true,
Locate an Op Supp Data record for the location and quarter where ParameterCode = "OPHOURS" and FuelCd is null.

If not found,
set ***Annual OPHOURS Calculated Value*** to null.
set ***OS OPHOURS Calculated Value*** to null
return result B

Otherwise,
if ***Annual Reporting Requirement*** == true
add OpValue to ***Annual OPHOURS Calculated Value.***

if this quarter is equal to 3 AND ***OS Reporting Requirement*** == true
add OpValue to ***OS OPHOURS Calculated Value.***

Otherwise,
set ***Annual OPHOURS Calculated Value*** to null.
set ***OS OPHOURS Calculated Value*** to null.

if (***Current Op Hours Summary Value Record*** is not null)

If (***Current Op Hours Summary Value Record.***YearToDateTotal is null and ***Annual Reporting Requirement*** == true) OR (***Current Op Hours Summary Value Record.***YearToDateTotal is less than 0,
append "YearToDateTotal" to ***OPHOURS Summary Invalid Fields***

If (***Current Op Hours Summary Value Record.***OzoneSeasonToDateTotal is null and ***OS Reporting Requirement*** == true AND Quarter of the ***Current Reporting Period*** is equal to 2 or 3 or 4), OR ***Current Op Hours Summary Value Record.***OzoneSeasonToDateTotal is less than 0,
append "OzoneSeasonToDateTotal" to ***OPHOURS Summary Invalid Fields***

If (***Current Op Hours Summary Value Record.***YearToDateTotal is not rounded to zero decimal places)
append "YearToDateTotal" to *Imprecise Fields*

If (***Current Op Hours Summary Value Record.***OzoneSeasonToDateTotal is not rounded to zero decimal places)
append "OzoneSeasonToDateTotal" to *Imprecise Fields*

If (***OPHOURS Summary Invalid Fields*** is not null)
return result C

else if (*Imprecise Fields* is not null)
set ***OPHOURS Summary Invalid Fields*** to *Imprecise Fields*
return result E

else if (***Annual OPHOURS Calculated Value*** is not null OR ***OS OPHOURS Calculated Value*** is not null)

Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "OPHOURS" AND
UOM = "HR"

if (***Annual OPHOURS Calculated Value*** is not null AND ***Annual OPHOURS Calculated Value*** <> ***Current Op Hours Summary Value Record.***YearToDateTotal)

append "YearToDateTotal" to ***OPTIME Summary Invalid Fields***

if (***OS OPHOURS Calculated Value*** is not null AND ***OS OPHOURS Calculated Value*** <> ***Current Op Hours Summary Value Record.***OzoneSeasonToDateTotal)

if (ABS(***OS OPHOURS Calculated Value - Current Op Hours Summary Value Record.***OzoneSeasonToDateTotal) > Tolerance OR quarter of the ***Current Reporting Period*** is greater than 2)

append "OzoneSeasonToDateTotal" to ***OPHOURS Summary Invalid Fields***

If ***OPHOURS Summary Invalid Fields*** is not null,

If (***OPHOURS Summary Invalid Fields*** contains "Year")

If (***OPHOURS Summary Invalid Fields*** contains "Ozone")

return result D

else

return result G

else

return result H

// if no result

if (***OS Reporting Requirement*** == false and ***Current Op Hours Summary Value Record.***OzoneSeasonToDateTotal is not null)

return result F

else if (***Annual Reporting Requirement*** == false and ***Current Op Hours Summary Value Record.***YearToDateTotal is not null)

return result I

Results:

Result	Response	Severity
A	The program could not determine ozone-season-to-date totals for [osparam], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
C	The [fieldname] in the Summary Value record for [param] is missing or invalid.	Critical Error Level 1
D	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc], and the OzoneSeasonToDateTotal of [osval] in the Summary Value record for [param] is inconsistent with the recalculated value of [oscalc].	Critical Error Level 1
E	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
F	You reported OzoneSeasonToDate in the Summary Value record for [param], but this is not valid for locations that are not associated with an ozone-season program.	Critical Error Level 1
G	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc].	Critical Error Level 1
H	The OzoneSeasonToDateTotal of [osval] in the Summary Value record for [param] is inconsistent with the recalculated value of [oscalc].	Critical Error Level 1
I	You reported YearToDate in the Summary Value record for [param], but this is not valid for locations that only report during the ozone season.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-17

Check Name: Check BCO2 Summary Value

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Current BCO2 Summary Value Record = SummaryValue record at this location where
Parameter = "BCO2" AND
Reporting Period ID = Current Reporting Period

Set ***RGGI Begin Date***, ***RGGI Start Quarter***, AND ***BCO2 Quarterly Reported Value*** to null.

if (***Current BCO2 Summary Value Record*** is not null)

if (***CurrentMonitorPlanLocationRecord***.StackPipeID is not null)
return result A

else

Locate a Program record for the unit where the ProgramCode == "RGGI", the UnitMonitorCertBeginDate is on or prior to the last day of the reporting period, and the EndDate is null or is on or after the first day of reporting period.

If not found,
return result B

else
Set ***RGGI Begin Date*** to the later of the UnitMonitorCertBeginDate and the EmissionsRecordingBeginDate (if not null) in the retrieved record.

If ***RGGI Begin Date*** is in a year prior to the current reporting period,
Set ***RGGI Start Quarter*** to 1
else
Set ***RGGI Start Quarter*** to the quarter of the ***RGGI Begin Date***.

if (***Current BCO2 Summary Value Record***.Current Reporting Period Total < 0)
return result C

else if (***Current BCO2 Summary Value Record***.Current Reporting Period Total is not rounded to one decimal place)
return result D

else
BCO2 Quarterly Reported Value = ***Current BCO2 Summary Value Record***.Current Reporting Period Total

Results:

Result	Response	Severity
A	You reported a Summary Value record for [param], but this value should only be reported at the unit, not at a stack or pipe.	Critical Error Level 1
B	You reported a Summary Value record for BCO2, but this location does not belong to the RGGI program during this reporting period.	Critical Error Level 1
C	The CurrentReportingPeriodTotal reported in the Summary Value record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
D	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
E	This check result is obsolete.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURAGG-18

Check Name: Compare BCO2 Mass YTD Values

Related Former Checks:

Applicability: General Check

Description:

Validation Tables:

[Quarterly Emissions Tolerances] (Cross Check Table)

Specifications:

Set ***Annual BCO2 Calculated Value*** to null.

If (***BCO2 Quarterly Reported Value*** is not null)
Annual BCO2M Calculated Value = ***BCO2 Quarterly Reported Value***
else
Annual BCO2M Calculated Value = -1

If (***RGGI Start Quarter*** is not null AND Quarter of the ***Current Reporting Period*** is greater than 1)

For each quarter in the current year from the ***RGGI Start Quarter*** to the quarter prior to the quarter of the ***Current Reporting Period***:

Locate an Op Supp Data record for the location and quarter where ParameterCode = "BCO2".

If not found,
If (***BCO2 Quarterly Reported Value*** is not null)
set ***Annual BCO2 Calculated Value*** to null
return result A

Otherwise,
If (***Annual BCO2M Calculated Value*** == -1)
Set ***Annual BCO2M Calculated Value*** to OpValue
else
add OpValue to ***Annual BCO2 Calculated Value.***

If (***Current BCO2 Summary Value Record*** is not null)

If (***Annual BCO2M Calculated Value*** = -1)
set ***Annual BCO2 Calculated Value*** to null
return result G

else if (***Current BCO2 Summary Value Record.***YearToDateTotal is null or is less than 0)
return result B

else if (***Current BCO2 Summary Value Record.***YearToDateTotal is not rounded to one decimal place)
return result C

else if (***Annual BCO2 Calculated Value*** is not null)

if (***Annual BCO2 Calculated Value*** <> ***Current BCO2 Summary Value Record.***YearToDateTotal)

Tolerance = Lookup Tolerance from Cross-Check Table "Quarterly Emissions Tolerances" where
Parameter = "CO2M" AND
UOM = "TON"

if (ABS(***Annual BCO2 Calculated Value - Current BCO2 Summary Value Record.***YearToDateTotal) >

Tolerance)
return result D

// If no result
If (***Current BCO2 Summary Value Record.***OzoneSeasonToDateTotal is not null)
return result E.

else
If (***Annual BCO2 Calculated Value*** == -1)
set ***Annual BCO2 Calculated Value*** to null

If (***Annual BCO2 Calculated Value*** > 0)
return result F

Results:

Result	Response	Severity
A	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The [fieldname] in the Summary Value record for [param] is missing or invalid.	Critical Error Level 1
C	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
D	The YearToDateTotal of [ytdval] in the Summary Value record for [param] is inconsistent with the recalculated value of [ytdcalc].	Critical Error Level 1
E	You reported OzoneSeasonToDate in the Summary Value record for [param], but this is not valid for this parameter.	Critical Error Level 1
F	You did not report a Summary Value record to report year-to-date total for [param].	Critical Error Level 1
G	You reported a Summary Value record for [param], but there was no [param] method defined in your monitoring plan that was active during the year.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Evaluation

Check Category:

Hourly Appendix D

Check Code: HOURAD-1

Check Name: Initialize Accumulators for Appendix D Calculations

Related Former Checks:

Applicability: Appendix D Check

Description: Set all Appendix D Accumulators to ZERO

Specifications:

HI App D Accumulator = 0
SO2 App D Accumulator = 0
CO2 App D Accumulator = 0
NOXR App E Accumulator = 0
Current Fuel Flow Record = null
Current Fuel Group = null
Fuels Used List = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Appendix D/E Unit-Level Initialization

Check Code: HOURAD-3

Check Name: Initialize Fuel Flow Record

Related Former Checks:

Applicability: Appendix D Check

Description: Initialization procedure for fuel flow category.

Specifications:

Current Fuel Group = ***Current Fuel Flow Record***.Fuel_Group_Cd

if (***Current Fuel Flow Record***.UnitFuelCd in set {OGS, PRG,OOL})
Special Fuel Burned = true

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-4

Check Name: Check Fuel Usage Time

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

HFF Usage Time Status = true

If (***Current Fuel Flow Record***.FuelUsageTime is null OR ***Current Fuel Flow Record***.FuelUsageTime < 0 OR ***Current Fuel Flow Record***.FuelUsageTime > 1)
    HFF Usage Time Status = false
    return result A
else if (***Current Hourly Op Record***.OperatingTime > 0 AND ***Current Hourly Op Record***.OperatingTime <= 1)

    if ***Current Fuel Flow Record***.FuelCode is not in ***Fuels Used List***
        add 1 to ***Fuel Op Hours Accumulator Array*** for the location and fuel
        append FuelCode to the ***Fuels Used List***

    if (***Current Fuel Flow Record***.FuelUsageTime > ***Current Hourly Op Record***.OperatingTime)
        HFF Usage Time Status = false
        return result B

    else if (***Hourly Fuel Flow Count For Gas + Hourly Fuel Flow Count For Oil*** == 1 AND (***MP Pipe Config for Hourly Checks*** is null OR ***Current Hourly Op Record***.LocationName begins with "CP") AND ***Current Fuel Flow Record***.FuelUsageTime <> ***Current Hourly Op Record***.OperatingTime)
        HFF Usage Time Status = false
        return result B

Results:

Result	Response	Severity
A	The FuelUsageTime reported in the HFF record for FuelCode [fuelcd] is invalid. The must be greater than 0 and less than or equal to 1.	Critical Error Level 1
B	The FuelUsageTime reported in the HFF record for FuelCode [fuelcd] is inconsistent with the OperatingTime for the hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-5

Check Name: Check Volumetric SODC Code

Related Former Checks:

Applicability: Appendix D Check

Description: Validation checks on Volumetric SODC Code for the Current Fuel Flow Record

Specifications:

HFF SODC Status = true
If (***Current Fuel Flow Record***.SourceOfDataVolumetricCode is null)
    If (***Current Fuel Flow Record***.VolumetricFlowRate is not null)
        HFF SODC Status == false
        return result A
else
    If (***Current Fuel Flow Record***.VolumetricFlowRate is null)
        HFF SODC Status == false
        return result B
    else if (***Current Fuel Group*** == "GAS" and ***Current Fuel Flow Record***.SourceOfDataVolumetricCode in set {5, 6})
        HFF SODC Status == false
        return result C
    else if (***Current Fuel Flow Record***.SourceOfDataVolumetricCode = "3" AND ***Current Unit Is Peaking*** == false)
        HFF SODC Status == false
        return result D
    else if (***HFF Fuel Indicator Code*** is not null)
        if (***Current Fuel Flow Record***.SourceOfDataVolumetricCode == "4" AND ***HFF Fuel Indicator Code*** <> "E")
            HFF SODC Status == false
            return result E
        else if (***Current Fuel Flow Record***.SourceOfDataVolumetricCode in set {5, 6} AND ***HFF Fuel Indicator Code*** <> "I")
            HFF SODC Status == false
            return result F

Results:

Result	Response	Severity
A	You did not report a [fieldname] in the HFF record for FuelCode [fuelcd], but you reported a [ratefieldname].	Critical Error Level 1
B	You reported a [fieldname] in the HFF record for FuelCode [fuelcd], but you did not report a [ratefieldname].	Critical Error Level 1
C	The SourceOfDataVolumetricCode reported in the HFF record for FuelCode [fuelcd] is invalid.	Critical Error Level 1
D	You reported a [fieldname] of 3 in the HFF record for FuelCode [fuelcd], but, according to the qualification record in your monitoring plan, this is not a peaking unit.	Critical Error Level 1
E	You reported a [fieldname] of 4 in the HFF record for FuelCode [fuelcd], which indicates that the fuel is an emergency fuel, but according to the Unit Fuel record in your monitoring plan, this fuel is not an emergency fuel.	Critical Error Level 1
F	You reported a [fieldname] of [sodc] in the HFF record for FuelCode [fuelcd], which indicates that the fuel is an igniter fuel, but according to the Unit Fuel record in your monitoring plan, this fuel is not an igniter fuel.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-6

Check Name: Check Oil Mass SODC Code

Related Former Checks:

Applicability: Appendix D Check

Description: Validation checks on Mass SODC Code for the Current Oil Fuel Flow Record

Specifications:

HFF Mass SODC Status == true
If (***Current Fuel Flow Record***.SourceOfDataMassCode is null)
    If (***Current Fuel Flow Record***.MassFlowRate is not null)
        HFF Mass SODC Status == false
        return result A
else
    If (***Current Fuel Flow Record***.MassFlowRate is null)
        HFF Mass SODC Status == false
        return result B
    else if (***Current Fuel Flow Record***.VolumetricFlowRate is not null AND ***Current Fuel Flow Record***.SourceOfDataMassCode <> "2")
        HFF Mass SODC Status == false
        return result C
    else if (***Current Fuel Flow Record***.VolumetricFlowRate is null AND ***Current Fuel Flow Record***.SourceOfDataMassCode == "2")
        HFF Mass SODC Status == false
        return result D
    else if (***Current Fuel Flow Record***.SourceOfDataMassCode = "3" AND ***Current Unit Is Peaking*** == false)
        HFF Mass SODC Status == false
        return result E
    else if (***HFF Fuel Indicator Code*** is not null)
        if (***Current Fuel Flow Record***.SourceOfDataMassCode == "4" AND ***HFF Fuel Indicator Code*** <> "E")
            HFF Mass SODC Status == false
            return result F
        else if (***Current Fuel Flow Record***.SourceOfDataMassCode in set {5, 6} AND ***HFF Fuel Indicator Code*** <> "I")
            HFF Mass SODC Status == false
            return result G

Results:

Result	Response	Severity
A	You did not report a [fieldname] in the HFF record for FuelCode [fuelcd], but you reported a [ratefieldname].	Critical Error Level 1
B	You reported a [fieldname] in the HFF record for FuelCode [fuelcd], but you did not report a [ratefieldname].	Critical Error Level 1
C	You reported a SourceOfDataMassCode of [sodc] in the HFF record for FuelCode [fuelcd], but you also reported a VolumetricFlowRate. The SourceOfDataMassCode must be 2 when mass oil flow is calculated from volumetric oil flow.	Critical Error Level 1
D	You reported a SourceOfDataMassCode of 2 in the HFF record for FuelCode [fuelcd], but you did not report a VolumetricFlowRate. The SourceOfDataMassCode should be 2 only when the mass oil rate is calculated from volumetric oil flow.	Critical Error Level 1
E	You reported a [fieldname] of 3 in the HFF record for FuelCode [fuelcd], but, according to the qualification record in your monitoring plan, this is not a peaking unit.	Critical Error Level 1
F	You reported a [fieldname] of 4 in the HFF record for FuelCode [fuelcd], which indicates that the fuel is an emergency fuel, but according to the Unit Fuel record in your monitoring plan, this fuel is not an emergency fuel.	Critical Error Level 1
G	You reported a [fieldname] of [sodc] in the HFF record for FuelCode [fuelcd], which indicates that the fuel is an igniter fuel, but according to the Unit Fuel record in your monitoring plan, this fuel is not an igniter fuel.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-7

Check Name: Check Fuel Flow Monitoring System

Related Former Checks:

Applicability: Appendix D Check

Description: Validates the Monitoring System reported in the HourlyFuelFlowData record

Specifications:

HFF System Type = null
FuelFlowComponentRecords = null
CurrentAppendixDStatus = null

If (***Current Fuel Flow Record***.MonitoringSystemID is null)
If (***Current Fuel Flow Record***.SourceOfDataVolumetricCode in set {0, 9} OR (***Current Fuel Group*** == "OIL" AND ***Current Fuel Flow Record***.SourceOfDataMassCode in set {0, 9}))
return result A
else if (***Legacy Data Evaluation*** == false AND (***Current Fuel Flow Record***.SourceOfDataVolumetricCode in set {1, 3} OR (***Current Fuel Group*** == "OIL" AND ***Current Fuel Flow Record***.SourceOfDataMassCode in set {1, 3})))
return result B
else if (***Current Fuel Group*** == "GAS")
HFF System Type = "GAS"
else if (***Current Fuel Flow Record***.VolumetricFlowRate is not null)
HFF System Type = "OILV"
else
HFF System Type = "OILM"
else
if (***Current Fuel Flow Record***.SourceOfDataVolumetricCode == "4")
return result C
else if (***Current Fuel Group*** == "OIL" AND (***Current Fuel Flow Record***.SourceOfDataVolumetricCode in set {5, 6} OR ***Current Oil Fuel Flow Record***.SourceOfDataMassCode in set {5, 6}))
return result C
else
Current Mon Sys Record = find active MonitoringSystem record where
MonitoringSystemId = ***Current Fuel Flow Record***.MonitoringSystemID

if *Current Mon Sys Record* is null
return result D
else if (***Current Fuel Group*** == "GAS" AND *Current Mon Sys Record*.SystemTypeCode <> "GAS")
return result E
else if (***Current Fuel Group*** == "OIL" AND *Current Mon Sys Record*.SystemTypeCode not in set {OILV, OILM})
return result F
else if (***Current Fuel Group*** == "OIL" AND ***Current Oil Fuel Flow Record***.SourceOfDataMassCode == "2" AND *Current Mon Sys Record*.SystemTypeCode <> "OILV")
return result G
else
HFF System Type = *Current Mon Sys Record*.SystemTypeCode
if *Current Mon Sys Record*.FuelCode is not null and is not equal to ***Current Fuel Flow Record***.FuelCode
HFF System Fuel = *Current Mon Sys Record*.FuelCode
return result H

else if (***Current Fuel Flow Record***.SourceOfDataVolumetricCode in set (0,9} OR (***Current Fuel Group*** == "OIL" AND ***Current Fuel Flow Record***.SourceOfDataMassCode in set {0,9}))

if (***Current Fuel Group*** = "OIL")

Locate ***MonitorSystemComponentRecordsByHourLocation*** where the SystemID is equal to ***CurrentFuelFlowRecord***.SystemID and the ComponentTypeCd = "OFFM" or "BOFF"

For each retrieved record found:

If (***MonitorSystemComponentRecordsByHourLocation***.ComponentTypeCd == "OFFM")

Add the ***MonitorSystemComponentRecordsByHourLocation*** record to ***FuelFlowComponentRecords***.

If none were found,
return result I.

else if (***Current Fuel Group*** = "GAS")

Locate ***MonitorSystemComponentRecordsByHourLocation*** where the SystemID is equal to ***CurrentFuelFlowRecord***.SystemID and the ComponentTypeCd = "GFFM" or "BGFF"

For each retrieved record found:

If (***MonitorSystemComponentRecordsByHourLocation***.ComponentTypeCd == "GFFM")

Add the ***MonitorSystemComponentRecordsByHourLocation*** record to ***FuelFlowComponentRecords***.

If none were found,
return result I.

Results:

Result	Response	Severity
A	You reported a SourceOfDataVolumetricCode or SourceOfDataMassCode in the HFF record for FuelCode [fuelcd], indicating the use of a fuel flowmeter system, but you did not report its MonitoringSystemID.	Critical Error Level 1
B	You reported a SourceOfDataVolumetricCode or SourceOfDataMassCode in the HFF record for FuelCode [fuelcd] that indicates the use of substitute data, but you did not report a MonitoringSystemID. This was not required for legacy EDR data, but for ECMPS, you should report the primary MonitoringSystemID of the fuel flowmeter system that normally records the flow for this fuel.	Critical Error Level 1
C	You reported a SourceOfDataVolumetricCode or SourceOfDataMassCode in the HFF record for FuelCode [fuelcd] that indicates the use of an emergency or igniter fuel, so you should not have reported a MonitoringSystemID in this record.	Critical Error Level 1
D	You reported MonitoringSystemID [ID] in the HFF record for FuelCode [fuelcd], but there is no MonitorSystem record for this system in your monitoring plan that was active during the hour.	Critical Error Level 1
E	You reported MonitoringSystemID [ID] in the HFF record for FuelCode [fuelcd], but this system is not a GAS monitoring system.	Critical Error Level 1
F	You reported MonitoringSystemID [ID] in the HFF record for FuelCode [fuelcd], but this system is not an OILM or OILV monitoring system.	Critical Error Level 1
G	You reported a SourceOfDataMassCode of 2 in the HFF record for FuelCode [fuelcd], indicating that you are calculating mass oil rate from volumetric oil flow, but MonitoringSystemID [ID] is not an OILV monitoring system.	Critical Error Level 1
H	Your reported MonitoringSystemID [ID] in the HFF record for FuelCode [fuelcd], but the FuelCode for this system in the MonitorSystem record is [sysfuel]. The FuelCode in the Monitor System record should be the same as the FuelCode in the HFF record.	Critical Error Level 1
I	You did not report any active fuel flowmeter components in your monitoring plan for MonitoringSystemID [ID]. The QA status for Appendix D testing for this system will not be evaluated.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-8

Check Name: Check Volumetric Units of Measure

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

HFF UOM Status = true
If (***Current Fuel Flow Record***.VolumetricUnitsOfMeasureCode is null)
    If (***Current Fuel Flow Record***.VolumetricFlowRate is not null)
        HFF UOM Status = false
        return result A
else
    If (***Current Fuel Flow Record***.VolumetricFlowRate is null)
        HFF UOM Status = false
        return result B
    else if (***Current Fuel Group*** == "OIL" AND ***Current Fuel Flow Record.***VolumetricUnitsOfMeasureCode is not in set {"GALHR", "BBLHR", "M3HR", "SCFH"})
        HFF UOM Status = false
        return result C
    else if (***Current Fuel Group*** == "GAS" AND ***Current Fuel Flow Record.***VolumetricUnitsOfMeasureCode <> "HSCF")
        HFF UOM Status = false
        return result C

Results:

Result	Response	Severity
A	You did not report a [fieldname] in the HFF record for FuelCode [fuelcd], but you reported a [ratefieldname].	Critical Error Level 1
B	You reported a [fieldname] in the HFF record for FuelCode [fuelcd], but you did not report a [ratefieldname].	Critical Error Level 1
C	The VolumetricUnitsOfMeasureCode reported in the HFF record for FuelCode [fuelcd] is invalid.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-9

Check Name: Check Fuel in HFF Record

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

HFF Fuel Indicator Code = null
Locate active UnitFuel record for the location
where FuelCd = ***Current Fuel Flow Record***.UnitFuelCd

If found,
HFF Fuel Indicator Code = ***Current Fuel Flow Record***.IndicatorCd
else
return result A

Results:

Result	Response	Severity
A	You did not report an active Unit Fuel record for FuelCode [fuelcd] in your monitoring plan.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-10

Check Name: Check Volumetric Flow in HFF Record

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

HFF Calc Volumetric Rate = null
HFF Max Heat Input for Volume = null

If (***HFF System Type*** is not null AND ***HFF SOD Status*** == true AND ***HFF Mass SODC Status*** == true AND ***HFF UOM Status*** == true)
If (***Current Fuel Flow Record***.VolumetricFlowRate is null)
If (***HFF System Type*** <> "OILM")
return result A
else
If (***HFF System Type*** == "OILM")
return result B
else if (***Current Fuel Flow Record***.SourceOfDataVolumetricCode == "4")

If (***Current Entity Type*** is equal to "Unit")
Locate a Unit Capacity record for the location and hour.

If exactly one record is found, and the MaximumHourlyHeatInputCapacity in the retrieved record is greater than 0,
HFF Max Heat Input for Volume =
UnitCapacityByHourLocation.MaximumHourlyHeatInputCapacity

If (***HFF GCV*** is not null)

If (***Current Fuel Flow Record***.MassFlowRate is null)

HFF Calc Volumetric Rate = HFF Max Heat Input for Volume / ***HFF GCV*** * 1000000, rounded to one decimal place.

else if (***HFF Density*** is not null)

HFF Calc Volumetric Rate = HFF Max Heat Input for Volume / ***HFF GCV*** / ***HFF Density*** * 1000000, rounded to one decimal place.

else
return result M

else
If (***Current Fuel Flow Record***.VolumetricFlowRate <= 0)
return result E
else
HFF Calc Volumetric Rate = ***Current Fuel Flow Record***.VolumetricFlowRate

else if (***Current Fuel Flow Record***.SourceOfDataVolumetricCode == "9"
If (***Current Fuel Group*** == "GAS")
HFF Volumetric Default Parameter = "MNGF"
else
HFF Volumetric Default Parameter = "MNOF"

Count active *Default Record* for the location where
ParameterCode == ***HFF Volumetric Default Parameter***

```
                FuelCode == Current Fuel Flow Record.FuelCode
        if (Count <> 1)
                return result C
        else if (Default Record.DefaultValue <= 0)
                return result D
        else if (Default Record.DefaultUnitsOfMeasureCode  == Current Fuel Flow
        Record.VolumetricUnitsOfMeasureCode)
                HFF Calc Volumetric Rate = Default Record.DefaultValue
                If (Current Fuel Flow Record.VolumetricFlowRate <= 0)
                        return result E
                else if (Current Fuel Flow Record.VolumetricFlowRate <> HFF Calc Volumetric Rate)
                        return result F
        else
                If (Current Fuel Flow Record.VolumetricFlowRate <= 0)
                        return result E
                else
                        return result G
else if Current Fuel Flow Record.MonitoringSystemID is not null)

        If (Current Fuel Flow Record.VolumetricFlowRate <= 0)
                return result E
        else
                If (Current Fuel Flow Record.SourceOfDataVolumetricCode <> 3)
                        HFF Calc Volumetric Rate = Current Fuel Flow Record.VolumetricFlowRate

                Count active System Fuel Flow Record for the system.

                If (Count <> 1)
                        return result H
                else if (System Fuel Flow Record.MaximumFuelFlowRate <= 0)
                        return result I
                else if (System Fuel Flow Record.SystemFuelFlowUOMCode  == Current Fuel Flow
                Record.VolumetricUnitsOfMeasureCode)
                        If (Current Fuel Flow Record.SourceOfDataVolumetricCode == 3)
                                HFF Calc Volumetric Rate = System Fuel Flow Record.MaximumFuelFlowRate
                                If Current Fuel Flow Record.VolumetricFlowRate <> HFF Calc Volumetric Rate)
                                        return result J
                        else
                                If (HFF Calc Volumetric Rate > System Fuel Flow Record.MaximumFuelFlowRate
                                        return result K

                else
                        return result L
else
        If (Current Fuel Flow Record.VolumetricFlowRate <= 0)
                return result E
        else
                HFF Calc Volumetric Rate = Current Fuel Flow Record.VolumetricFlowRate
```

Results:

Result	Response	Severity
A	You did not report a VolumetricFlowRate in the HFF record for FuelCode [fuelcd], which is required when using [systype] MonitoringSystemID [ID].	Critical Error Level 1
B	You reported a VolumetricFlowRate in the HFF record for FuelCode [fuelcd], which is invalid when using an OILM system.	Critical Error Level 1
C	You did not report one and only one default record for [parameter] for FuelCode [fuelcd] in your monitoring plan that was active during current hour.	Critical Error Level 1
D	The DefaultValue reported in the active [parameter] default record for the hour is invalid.	Critical Error Level 1
E	The VolumetricFlowRate reported in the HFF record for FuelCode [fuelcd] is invalid.	Critical Error Level 1
F	You reported a SourceOfDataVolumetricCode of [sodc] in the HFF record for FuelCode [fuelcd], but the VolumetricFlowRate is not equal to the fuel flow rate defined in the active [parameter] default record in your monitoring plan.	Critical Error Level 1
G	The VolumetricUnitsOfMeasureCode in the HFF record for FuelCode [fuelcd] is not the same as the DefaultUnitsOfMeasureCode in the active [parameter] default record in your monitoring plan.	Critical Error Level 1
H	You did not report one and only one active SystemFuelFlow record for MonitoringSystemID [ID] in your monitoring plan for the hour.	Critical Error Level 1
I	The MaximumFuelFlowRate reported in the active System Fuel Flow record for MonitoringSystemID [ID] in your monitoring plan is invalid.	Critical Error Level 1
J	You reported a SourceOfDataVolumetricCode of 3 in the HFF record for FuelCode [fuelcd], but the VolumetricFlowRate is not equal to the MaximumFuelFlowRate specified in the active System Fuel Flow record for MonitoringSystemID [ID] in your monitoring plan.	Critical Error Level 1
K	Warning: The VolumetricFlowRate reported in the HFF record for FuelCode [fuelcd] exceeds the MaximumFuelFlowRate specified in the active System Fuel Flow record for MonitoringSystemID [ID] in your monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. You should investigate the cause of these exceedances and determine whether an adjustment to your monitoring systems or monitoring plan is necessary.	Informational Message
L	The VolumetricUnitsOfMeasureCode in the HFF record for FuelCode [fuelcd] is not the same as the SystemFuelFlowUOMCode in the active System Fuel Flow record for MonitoringSystemID [ID] in your monitoring plan.	Critical Error Level 1
M	You did not report one and only one valid active Unit Capacity record in your monitoring plan for the unit for the hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-11

Check Name: Check Mass Oil Flow in HFF Record

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

HFF Calc Mass Oil Rate = null
HFF Max Heat Input for Mass = null
If (***HFF System Type*** is not null AND ***HFF SOD Status*** == true AND ***HFF Mass SODC Status*** == true AND ***HFF UOM Status*** == true)
    If (***Current Fuel Flow Record***.MassFlowRate is null)
        If (***HFF System Type*** == "OILM")
            return result A
        Else if (***HFF System Type*** == "OILV"
            If (***Current Fuel Flow Record***.SourceOfDataMassCode == "2")
                return result B
            Else if (***Current Unit is ARP*** == true)
                return result C
    else
        If (***HFF System Type*** == "GAS")
            return result D
        else if (***HFF System Type*** == "OILV")
            If (***Current Fuel Flow Record***.SourceOfDataMassCode == "2" AND ***Current Fuel Flow Record***.MassFlowRate <= 0)
                return result E
        else if (***Current Fuel Flow Record***.SourceOfDataMassCode == "4")

            If (***Current Entity Type*** is equal to "Unit")
                Locate a Unit Capacity record for the location and hour.

                If exactly one record is found, and the MaximumHourlyHeatInputCapacity in the retrieved record is greater than 0,
                    HFF Max Heat Input for Mass = ***UnitCapacityByHourLocation***.MaximumHourlyHeatInputCapacity

                    If (***HFF GCV*** is not null)

                        ***HFF Calc Mass Oil Rate = HFF Max Heat Input for Mass / HFF GCV** * 1000000, rounded to one decimal place.

                else
                    return result M
            else
                If (***Current Fuel Flow Record***.MassFlowRate <= 0)
                    return result E
                else
                    HFF Calc Mass Oil Rate = ***Current Fuel Flow Record***.MassFlowRate

        else if (***Current Fuel Flow Record***.SourceOfDataMassCode == "9")
            HFF Mass Oil Default Parameter = "MNOF"

            Count active *Default Record* for the location where
                ParameterCode == ***HFF Mass Oil Default Parameter***
                FuelCode == ***Current Fuel Flow Record***.FuelCode
            if (*Count* <> 1)
                return result F

```
            else if (Default Record.DefaultValue <= 0 OR Default Record.DefaultUnitsOfMeasureCode  <> "LBHR")
                  return result G
            else
                  HFF Calc Mass Oil Rate = Default Record.DefaultValue
                  If (Current Fuel Flow Record.MassFlowRate <= 0)
                        return result E
                  else if (Current Fuel Flow Record.MassFlowRate <> HFF Calc Mass Oil Rate)
                        return result H

else if (Current Fuel Flow Record.MonitoringSystemID is not null)

      If (Current Fuel Flow Record.MassFlowRate <= 0)
            return result E
      else
            If (Current Fuel Flow Record.SourceOfDataMassCode <> 3)
                  HFF Calc Mass Oil Rate = Current Fuel Flow Record.MassFlowRate

            Count active System Fuel Flow Record for the system.

            If (Count <> 1)
                  return result I

            else if (System Fuel Flow Record.MaximumFuelFlowRate <= 0 OR System Fuel Flow
            Record.SystemFuelFlowUOMCode  <> "LBHR")
                  return result J
            else
                  If (Current Fuel Flow Record.SourceOfDataMassCode == 3)
                        HFF Calc Mass Oil Rate = System Fuel Flow Record.MaximumFuelFlowRate
                        If (Current Fuel Flow Record.MassFlowRate <> HFF Calc Mass Oil Rate)
                              return result K
                  else
                        If (HFF Calc Mass Oil Rate > System Fuel Flow Record.MaximumFuelFlowRate
                              return result L
else
      If (Current Fuel Flow Record.MassFlowRate <= 0)
            return result E
      else
            HFF Calc Mass Oil Rate = Current Fuel Flow Record.MassFlowRate
```

Results:

Result	Response	Severity
A	You did not report a [fieldname] in the HFF record for FuelCode [fuelcd], but the MonitoringSystemID [ID] is an [systype] fuel flow system.	Critical Error Level 1
B	You reported a SourceOfDataMassCode of 2 in the HFF record for FuelCode [fuelcd], which indicates that the mass oil rate was calculated from the volumetric oil rate, but you did not report a MassFlowRate in the record.	Critical Error Level 1
C	You did not report a [fieldname] in the HFF record for FuelCode [fuelcd], but this value is required for an ARP unit.	Critical Error Level 1
D	You reported a MassFlowRate in the HFF record for FuelCode [fuelcd]. This value should be blank for a gas fuel.	Critical Error Level 1
E	The MassFlowRate reported in the HFF record for FuelCode [fuelcd] is invalid.	Critical Error Level 1
F	You did not report one and only one default record for [parameter] for FuelCode [fuelcd] in your monitoring plan that was active during current hour.	Critical Error Level 1
G	The DefaultValue or DefaultUnitsOfMeasureCode reported in the active [parameter] default record for the hour is invalid.	Critical Error Level 1
H	You reported a SourceOfDataMassCode of [sodc] in the HFF record for FuelCode [fuelcd], but the MassFlowRate is not equal to the fuel flow rate defined in the active [parameter] default record in your monitoring plan.	Critical Error Level 1
I	You did not report one and only one active SystemFuelFlow record for MonitoringSystemID [ID] in your monitoring plan for the hour.	Critical Error Level 1
J	The MaximumFuelFlowRate or SystemFuelFlowUOMCode reported in the active System Fuel Flow record for MonitoringSystemID [ID] in your monitoring plan is invalid.	Critical Error Level 1
K	You reported a SourceOfDataMassCode of 3 in the HFF record for FuelCode [fuelcd], but the MassFlowRate is not equal to the MaximumFuelFlowRate specified in the active System Fuel Flow record for MonitoringSystemID [ID] in your monitoring plan.	Critical Error Level 1
L	Warning: The MassFlowRate reported in the HFF record for FuelCode [fuelcd] exceeds the MaximumFuelFlowRate specified in the active System Fuel Flow record for MonitoringSystemID [ID] in your monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. You should investigate the cause of these exceedances and determine whether an adjustment to your monitoring systems or monitoring plan is necessary.	Informational Message
M	You did not report one and only one valid active Unit Capacity record in your monitoring plan for the unit for the hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-12

Check Name: Determine Density

Related Former Checks:

Applicability: Appendix D Check

Description:

Validation Tables:

Fuel Type Reality Checks for Density (Cross Check Table)
Fuel Type Warning Levels for Density (Cross Check Table)
Table D-6 Missing Data Values (Cross Check Table)

Specifications:

HFF Density = null
Current Density Record = null

Count the HourlyParamFuelFlow record where
HourlyParamFuelFlow.HourlyFuelFlowID = ***Current Fuel Flow Record***.HourlyFuelFlowID AND
HourlyParamFuelFlow.ParameterCode = "DENSOIL"

If (*Count* > 1)
return result A
Else If (*Count* == 0)
If (***HFF System Type*** == "OILV" and ***Current Fuel Flow Record***.SourceOfDataMassCode == "2")
return result B
Else if (***HFF System Type*** = "OILV" AND ***Current Fuel Flow Record***.SourceOfDataMassCode == "2")
Current Density Record = matching record
Density UOM = ***Current Density Record***.ParameterUOMCode

if (*Density UOM* not in set {LBGAL, LBBBL, LBM3, LBSCF})
return result C
else if (***Current Fuel Flow Record***.VolumetricUnitsOfMeasureCode == "GALHR" AND *Density UOM* <> "LBGAL")
return result D
else if (***Current Fuel Flow Record***.VolumetricUnitsOfMeasureCode == "BBLHR" AND *Density UOM* <> "LBBBL")
return result D
else if (***Current Fuel Flow Record***.VolumetricUnitsOfMeasureCode == "M3HR" AND *Density UOM* <> "LBM3")
return result D
else if (***Current Fuel Flow Record***.VolumetricUnitsOfMeasureCode == "SCFH" AND *Density UOM* <> "LBSCF")
return result D
else if (***Current Density Record***.ParamValFuel > 0)

Density Default = null
If (***Current Density Record***.SampleTypeCode == 8)
Density Default = Lookup "MissingDataValue" in "Table D-6 Missing Data Values"
where "Parameter" column = "DENSOIL - " + *Density UOM* AND "FuelCode" column = ***Current Fuel Flow Record***.FuelCode

If (*Density Default* == null)

Max Expected Density = Lookup "Upper Value" in "Fuel Type Warning Levels for Density Cross Check Table"
where "Fuel Code - Units Of Measure" column = concatenation of (***Current Fuel Flow Record***.FuelCode, " - ", *Density UOM*)
Min Expected Density = Lookup "Lower Value" in "Fuel Type Warning Levels for Density Cross Check Table"
where "Fuel Code - Units Of Measure" column = concatenation of (***Current Fuel Flow Record***.FuelCode, " - ", *Density UOM*)
Max Allowed Density = Lookup "Upper Value" in "Fuel Type Reality Checks for Density Cross Check Table"
where "Fuel Code - Units Of Measure" column = concatenation of (***Current Fuel Flow Record***.FuelCode,

```
                    " - ", Density UOM)
                Min Allowed Density = Lookup "Lower Value" in "Fuel Type Reality Checks for Density Cross Check Table"
                    where "Fuel Code - Units Of Measure" column = concatenation of (Current Fuel Flow Record.FuelCode,
                    " - ", Density UOM)

                if (Max Allowed Density is not null AND Current Density Record.ParamValFuel > Max Allowed Density) OR
                (Min Allowed Density is not null AND Current Density Record.ParamValFuel < Min Allowed Density)
                    return result E
                else
                    HFF Density = Current Density Record.ParamValFuel
                    if (Min Expected Density is not null AND HFF Density  < Min Expected Density) OR (Max Expected
                    Density is not null AND HFF Density > Max Expected Density)
                        return result F

            else
                if (Density Default == Current Density Record.ParamValFuel)
                    HFF Density = Current Density Record.ParamValFuel
                else
                    return result G
        else
            return result H
else
        return result I
```

Results:

Result	Response	Severity
A	You reported more than one HPFF record for [parameter] for FuelCode [fuelcd] for the hour.	Critical Error Level 1
B	You did not report an HPFF record for [parameter] for FuelCode [fuelcd] for the hour.	Critical Error Level 1
C	The ParameterUOMCode reported in the HPFF record for DENSOIL for FuelCode [fuelcd] is missing or invalid.	Critical Error Level 1
D	The ParameterUOMCode reported in the HPFF record for DENSOIL for FuelCode [fuelcd] is inconsistent with the VolumetricUnitsOfMeasureCode reported in the associated HFF record.	Critical Error Level 1
E	The ParameterValueForFuel reported in the HPFF record for [parameter] for FuelCode [fuelcd] is outside the range of allowable values for the fuel type.	Critical Error Level 1
F	The ParameterValueForFuel reported in the HPFF record for [parameter] for FuelCode [fuelcd] is outside the range of expected values for the fuel type.	Critical Error Level 2
G	You reported a SampleTypeCode of 8 in the HPFF record for [parameter] for FuelCode [fuelcd], indicating the use of a Table D-6 default, but the ParameterValueForFuel does not equal the default value for the fuel.	Critical Error Level 1
H	The ParameterValueForFuel reported in the HPFF record for [parameter] for FuelCode [fuelcd] is invalid. The value should be greater than 0.	Critical Error Level 1
I	You reported an HPFF record for [parameter] for FuelCode [fuelcd], but this value is only appropriate when using an OILV system and a SourceOfDataMassCode equal to 2.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-13

Check Name: Check Density Sample Type

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current Density Record*** is not null)
If ***Current Density Record***.SampleTypeCode not in {1, 2, 5, 6, 7, 8}
return result A

Results:

Result	Response	Severity
A	The SampleTypeCode reported in the HPFF record for DENSOIL for FuelCode [fuelcd] is missing or invalid.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-14

Check Name: Check Extraneous Density Record Fields

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

Hourly Extraneous Fields = null

If (***Current Density Record*** is not null)

    If (***Current Density Record***.FormulaIdentifier is not null)
        append "FormulaIdentifier" to ***Hourly Extraneous Fields***
    If (***Current Density Record***.MonitoringSystemID is not null)
        append "MonitoringSystemID" to ***Hourly Extraneous Fields***
    If (***Current Density Record***.SegmentNumber is not null)
        append "SegmentNumber" to ***Hourly Extraneous Fields***
    If (***Current Density Record***.OperatingConditionCode is not null)
        append "OperatingConditionCode" to ***Hourly Extraneous Fields***

    If (***Hourly Extraneous Fields*** is not null)
        return result A

Results:

Result	Response	Severity
A	You reported [fieldnames] in the HPFF record for DENSOIL for FuelCode [fuelcd]. This data should be blank.	Non-Critical Error

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-15

Check Name: Calculate Mass Oil Flow

Related Former Checks:

Applicability: Appendix D Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

If HFF Calc Volumetric Flow is not null AND ***HFF Density*** is not null)

HFF Calc Mass Oil Flow = ***HFF Density * HFF Calc Volumetric Flow***, and round the result to one decimal place (0.1)

Flow Rate Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "OILM" AND
UOM = "LBHR"

If (***Current Fuel Flow Record***.MassFlowRate > 0)
if (ABS(***Current Fuel Flow Record***.MassFlowRate - ***HFF Calc Mass Oil Flow***) > Flow Rate Tolerance)
return result A

Results:

Result	Response	Severity
A	The MassFlowRate reported in the HFF record for FuelCode [fuelcd] is inconsistent with the value calculated from the VolumetricFlowRate and density.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-16

Check Name: Determine GCV

Related Former Checks:

Applicability: Appendix D Check

Description:

Validation Tables:

Fuel Type Reality Checks for GCV (Cross Check Table)
Fuel Type Warning Levels for GCV (Cross Check Table)
Table D-6 Missing Data Values (Cross Check Table)

Specifications:

HFF GCV = null
Current GCV Record = null

Count the HourlyParamFuelFlow record where
HourlyParamFuelFlow.HourlyFuelFlowID = ***Current Fuel Flow Record***.HourlyFuelFlowID AND
HourlyParamFuelFlow.ParameterCode = "GCV"

If (*Count* > 1)
return result A
Else If (*Count* == 0)
If (***Current HI HPFF Record*** is not null)
return result B
Else if (***Current HI HPFF Record*** is not null)
Current GCV Record = matching record
GCV UOM = ***Current GCV Record***.ParameterUOMCode

if (*GCV UOM* not in set {BTUGAL, BTUBBL, BTUM3, BTUSCF, BTULB, BTUHSCF}
return result C
else if (***Current Fuel Group*** == "GAS" AND *GCV UOM* <> "BTUHSCF")
return result D
else if (***Current Fuel Group*** == "OIL" AND ***Current Fuel Flow Record***.MassFlowRate is not null AND *GCV UOM* <> "BTULB")
return result D
else if (***Current Fuel Group*** == "OIL" AND ***Current Fuel Flow Record***.MassFlowRate is null AND ***Current Fuel Flow Record***.VolumetricUnitsOfMeasureCode == "GALHR" AND *GCV UOM* <> "BTUGAL")
return result D
else if (***Current Fuel Group*** == "OIL" AND ***Current Fuel Flow Record***.MassFlowRate is null AND ***Current Fuel Flow Record***.VolumetricUnitsOfMeasureCode == "BBLHR" AND *GCV UOM* <> "BTUBBL")
return result D
else if (***Current Fuel Group*** == "OIL" AND ***Current Fuel Flow Record***.MassFlowRate is null AND ***Current Fuel Flow Record***.VolumetricUnitsOfMeasureCode == "M3HR" AND *GCV UOM* <> "BTUM3")
return result D
else if (***Current Fuel Group*** == "OIL" AND ***Current Fuel Flow Record***.MassFlowRate is null AND ***Current Fuel Flow Record***.VolumetricUnitsOfMeasureCode == "SCFH" AND *GCV UOM* <> "BTUSCF")
return result D
else if (***Current GCV Record***.ParamValFuel > 0)

GCV Default = null
If (***Current GCV Record***.SampleTypeCode == 8)
GCV Default = Lookup "MissingDataValue" in "Table D-6 Missing Data Values"
where "Parameter" column = "GCV - " + *GCV UOM* AND "FuelCode" column = ***Current Fuel Flow Record***.FuelCode

If (*GCV Default* == null)

GCV Default = Lookup "MissingDataValue" in "Table D-6 Missing Data Values"
where "Parameter" column = "GCV - " + *GCV UOM* and FuelCode column is null.

If (*GCV Default* == null)

Max Expected GCV = Lookup "Upper Value" in "Fuel Type Warning Levels for GCV Cross Check Table"
where "Fuel Code - Units Of Measure" column = concatenation of (***Current Fuel Flow Record***.FuelCode, " - ", *GCV UOM*)
Min Expected GCV = Lookup "Lower Value" in "Fuel Type Warning Levels for GCV Cross Check Table"
where "Fuel Code - Units Of Measure" column = concatenation of (***Current Fuel Flow Record***.FuelCode, " - ", *GCV UOM*)
Max Allowed GCV = Lookup "Upper Value" in "Fuel Type Reality Checks for GCV Cross Check Table"
where "Fuel Code - Units Of Measure" column = concatenation of (***Current Fuel Flow Record***.FuelCode, " - ", *GCV UOM*)
Min Allowed GCV = Lookup "Lower Value" in "Fuel Type Reality Checks for GCV Cross Check Table"
where "Fuel Code - Units Of Measure" column = concatenation of (***Current Fuel Flow Record***.FuelCode, " - ", *GCV UOM*)

if (*Max Allowed GCV* is not null AND ***Current GCV Record***.ParamValFuel > *Max Allowed GCV)* OR (*Min Allowed GCV* is not null AND ***Current GCV Record***.ParamValFuel < *Min Allowed GCV*)
return result E
else if (***Current GCV Record***.ParamValFuel is not rounded to one decimal place)
return result J
else
HFF GCV = ***Current GCV Record***.ParamValFuel
if (*Min Expected GCV* is not null AND ***HFF GCV*** < *Min Expected GCV)* OR (*Max Expected GCV* is not null AND ***HFF GCV*** > *Max Expected GCV*)
return result F

else
If (***Current GCV Record***.ParamValFuel is not rounded to one decimal place)
return result J
else if (GCV Default == ***Current GCV Record***.ParamValFuel)
HFF GCV = ***Current GCV Record***.ParamValFuel
else
return result G
else
return result H
else
return result I

Results:

Result	Response	Severity
A	You reported more than one HPFF record for [parameter] for FuelCode [fuelcd] for the hour.	Critical Error Level 1
B	You did not report a HPFF record for GCV for FuelCode [fuelcd] for the hour.	Critical Error Level 1
C	The ParameterUOMCode reported in the HPFF record for GCV for FuelCode [fuelcd] is missing or invalid.	Critical Error Level 1
D	The ParameterUOMCode reported in the HPFF record for GCV for FuelCode [fuelcd] is inconsistent with the fuel flow units of measure.	Critical Error Level 1
E	The ParameterValueForFuel reported in the HPFF record for [parameter] for FuelCode [fuelcd] is outside the range of allowable values for the fuel type.	Critical Error Level 1
F	The ParameterValueForFuel reported in the HPFF record for [parameter] for FuelCode [fuelcd] is outside the range of expected values for the fuel type.	Non-Critical Error
G	You reported a SampleTypeCode of 8 in the HPFF record for [parameter] for FuelCode [fuelcd], indicating the use of a Table D-6 default, but the ParameterValueForFuel does not equal the default value for the fuel.	Critical Error Level 1
H	The ParameterValueForFuel reported in the HPFF record for GCV for FuelCode [fuelcd] is invalid. The value must be greater than 0.	Critical Error Level 1
I	You reported an HPFF record for GCV for FuelCode [fuelcd], but you have not reported an HPFF record for HI for the hour.	Critical Error Level 1
J	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-17

Check Name: Check GCV Sample Type

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current GCV Record*** is not null)
    If (***Current Fuel Group*** == "OIL" AND ***Current GCV Record***.SampleTypeCode not in {1, 2, 5, 6, 7, 8})
        return result A
    else if (***Current Fuel Group*** == "GAS" AND ***Current GCV Record***.SampleTypeCode not in {0, 2, 3, 4, 6, 7, 8})
        return result A

Results:

Result	Response	Severity
A	The SampleTypeCode reported in the HPFF record for GCV for FuelCode [fuelcd] is missing or invalid.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-18

Check Name: Check Extraneous GCV Record Fields

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

Hourly Extraneous Fields = null

If (***Current GCV Record*** is not null)

If (***Current GCV Record***.FormulaIdentifier is not null)
append "FormulaIdentifier" to ***Hourly Extraneous Fields***
If (***Current GCV Record***.MonitoringSystemID is not null)
append "MonitoringSystemID" to ***Hourly Extraneous Fields***
If (***Current GCV Record***.SegmentNumber is not null)
append "SegmentNumber" to ***Hourly Extraneous Fields***
If (***Current GCV Record***.OperatingConditionCode is not null)
append "OperatingConditionCode" to ***Hourly Extraneous Fields***

If (***Hourly Extraneous Fields*** is not null)
return result A

Results:

Result	Response	Severity
A	You reported [fieldnames] in the HPFF record for GCV for FuelCode [fuelcd]. This data should be blank.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-19

Check Name: Validate Heat Input Record

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

Current HI HPFF Record = null

Count the HourlyParamFuelFlow record where
HourlyParamFuelFlow.HourlyFuelFlowID = ***Current Fuel Flow Record***.HourlyFuelFlowID AND
HourlyParamFuelFlow.ParameterCode = "HI"

If (*Count* > 1)
HI App D Accumulator = -1
return result A
Else If (*Count* == 0)
If (***Heat Input App D Method Active For Hour*** == true)
HI App D Accumulator = -1
return result B
Else if (***Heat Input App D Method Active For Hour*** == true)
Current HI HPFF Record = matching record
HI HPFF Exists = true

if (***Current HI HPFF Record***.MonitoringFormulaId is null
return result C
else
Cur HI Mon Formula Record = Lookup active formula in MonitoringFormula Table where
MonitoringFormulaID = ***Current HI HPFF Record***.MonitoringFormulaID

if (*Cur HI Mon Formula Record* is null)
return result D
else if (*Cur HI Mon Formula Record*.ParameterCode <> "HI")
return result E
else if (***Current Fuel Group*** == "GAS")
If (*Cur HI Mon Formula Record*.EquationCode not in set {D-6, F-20})
return result F
else if (***Current Fuel Flow Record***.MassFlowRate is not null)
If (*Cur HI Mon Formula Record*.EquationCode not in set {D-8, F-19})
return result F
else
If (*Cur HI Mon Formula Record*.EquationCode <> "F-19V")
return result F
else
return result G

Results:

Result	Response	Severity
A	You reported more than one HPFF record for [parameter] for FuelCode [fuelcd] for the hour.	Critical Error Level 1
B	You did not report an HPFF record for [parameter] for FuelCode [fuelcd] for the hour.	Critical Error Level 1
C	You did not report a FormulaID in the HPFF record for HI for FuelCode [fuelcd].	Critical Error Level 1
D	You reported FormulaID [ID] in the HPFF record for HI for FuelCode [fuelcd], but there is no active Formula record for this formula in your monitoring plan.	Critical Error Level 1
E	You reported FormulaID [ID] in the HPFF record for HI for FuelCode [fuelcd], but this is not an HI formula.	Critical Error Level 1
F	The FormulaCode of FormulaID [ID] reported in the HPFF record for HI for FuelCode [fuelcd] is invalid.	Critical Error Level 1
G	You reported an HPFF record for GCV for FuelCode [fuelcd], but you have not reported an HPFF record for HI for the hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-20

Check Name: Check Extraneous Heat Input Record Fields

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

Hourly Extraneous Fields = null

If (***Current HI HPFF Record*** is not null)

If (***Current HI HPFF Record***.MonitoringSystemID is not null)
append "MonitoringSystemID" to ***Hourly Extraneous Fields***
If (***Current HI HPFF Record***.SegmentNumber is not null)
append "SegmentNumber" to ***Hourly Extraneous Fields***
If (***Current HI HPFF Record***.OperatingConditionCode is not null)
append "OperatingConditionCode" to ***Hourly Extraneous Fields***
If (***Current HI HPFF Record***.SampleTypeCode is not null)
append "SampleTypeCode" to ***Hourly Extraneous Fields***

If (***Hourly Extraneous Fields*** is not null)
return result A

Results:

Result	Response	Severity
A	You reported [fieldnames] in the HPFF record for HI for FuelCode [fuelcd]. This data should be blank.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-21

Check Name: Calculate Heat Input Rate

Related Former Checks:

Applicability: Appendix D Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

HFF Calc HI Rate = null

If (***Current HI HPFF Record*** is not null)

if (***Current Fuel Flow Record***.SourceOfDataVolumetricCode == "4")
if (***HFF Max Heat Input for Volume*** is not null)
HFF Calc HI Rate = ***HFF Max Heat Input for Volume***
else
HI App D Accumulator = -1
return result A

else if (***Current Fuel Flow Record***.SourceOfDataMassCode == "4")
if (***HFF Max Heat Input for Mass*** is not null)
HFF Calc HI Rate = ***HFF Max Heat Input for Mass***

else
HI App D Accumulator = -1
return result A

else if (***HFF GCV*** is not null)

HI HPFF Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "HI HPFF" AND
UOM = "MMBTUHR"

If (***HFF System Type*** == "GAS" OR ***Current Fuel Flow Record***.MassFlowRate is null)
If (***HFF Calc Volumetric Rate*** is not null)
HFF Calc HI Rate = ***HFF Calc Volumetric Rate*** * ***HFF GCV*** / 1000000, and round the result to one decimal place.
else
HI App D Accumulator = -1
return result A

else
if (***HFF Calc Mass Oil Rate*** is not null)
HFF Calc HI Rate = ***HFF Calc Mass Oil Rate*** * ***HFF GCV*** / 1000000, and round the result to one decimal place.
else
HI App D Accumulator = -1
return result A

If (***HFF Calc HI Rate*** is not null)
If ***Current Fuel Flow Record***.FuelUsageTime > 0 AND ***Current Fuel Flow Record***.FuelUsageTime <= 1 AND ***HI App D Accumulator*** >= 0)
// Note - this accumulates totals for all Fuel flow records and does not work like a normal parameter
HI App D Accumulator = ***HI App D Accumulator*** + ***HFF Calc HI Rate*** * ***Current Fuel Flow***

Record.FuelUsageTime
else
HI App D Accumulator = -1

If (***Current HI HPFF Record***.ParamValFuel > 0)

If (***Current Fuel Flow Record***.SourceOfDataVolumetricCode == "4" OR ***Current Fuel Flow Record***.SourceOfDataMassCode == "4")

if (HFF Calc HI Rate is equal to ***Current HI HPFF Record.***ParamValFuel)

if (***Current Fuel Flow Record***.SourceOfDataVolumetricCode == "4")

If (***Current Fuel Flow Record***.VolumetricFlowRate is greater than 0, AND ***HFF Calc Volumetric Rate*** is not equal to ***Current Fuel Flow Record***.VolumetricFlowRate)

Flow Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where Parameter = "FOIL"

if (ABS(***HFF Calc Volumetric Rate - Current Fuel Flow Record***.VolumetricFlowRate) > *Flow Tolerance*)
return result C

else

If (***Current Fuel Flow Record***.MassFlowRate is greater than 0, AND ***HFF Calc Mass Oil Rate*** is not equal to ***Current Fuel Flow Record***.MassFlowRate)

Flow Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where Parameter = "FOIL"

if (ABS(***HFF Calc Mass Oil Rate - Current Fuel Flow Record***.MassFlowRate) > *Flow Tolerance*)
return result D

else
If (ABS(***HFF Calc HI Rate - Current HI HPFF Record.***ParamValFuel) > *HI HPFF Tolerance*)
return result B
else
HI App D Accumulator = -1
return result A

Results:

Result	Response	Severity
A	The ParameterValueForFuel in the HPFF record for [parameter] for FuelCode [fuelcd] could not be recalculated due to errors listed above.	Informational Message
B	The ParameterValueForFuel reported in the HPFF record for HI for FuelCode [fuelcd] is inconsistent with the recalculated value.	Critical Error Level 1
C	You reported a SourceOfDataVolumetricCode of 4, indicating that you burned an emergency fuel, but the VolumetricFlowRate in the HFF record for FuelCode [fuelcd] is inconsistent with the maximum fuel flow rate for the unit. When you burn an emergency fuel, you should report the maximum fuel flow rate, which is based on the maximum hourly heat input capacity of the unit.	Critical Error Level 1
D	You reported a SourceOfDataMassCode of 4, indicating that you burned an emergency fuel, but the MassFlowRate in the HFF record for FuelCode [fuelcd] is inconsistent with the maximum fuel flow rate for the unit. When you burn an emergency fuel, you should report the maximum fuel flow rate, which is based on the maximum hourly heat input capacity of the unit.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-22

Check Name: Check Reported Heat Input

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current HI HPFF Record*** is not null)

If (***Current HI HPFF Record***.ParamValFuel >= 0
If (***Current HI HPFF Record***.ParamValFuel is not rounded to one decimal place)
return result D

else if (***Current Fuel Flow Record***.SourceOfDataVolumetricCode == "4" and ***HFF Max Heat Input for Volume*** is not null)
if (***Current HI HPFF Record.***ParamValFuel is not equal to ***HFF Max Heat Input for Volume***)
return result E

else if (***Current Fuel Flow Record***.SourceOfDataMassCode == "4" and ***HFF Max Heat Input for Mass*** is not null)
if (***Current HI HPFF Record.***ParamValFuel is not equal to ***HFF Max Heat Input for Mass***)
return result F

else
If ***Current Entity Type*** == "CP",
Count active UnitCapacity record for each unit linked to the pipe
if (*Count* <> 1 for any unit)
return result A
else
Calculate Max Heat Input as the sum of *Unit Capacity Record*.MaximumHourlyHeatInputCapacity for all units.
else
Count active UnitCapacity record for the associated unit.
if (*Count* <> 1)
return result A
else
Max Heat Input = *Unit Capacity Record*.MaximumHourlyHeatInputCapacity

if ***Current HI HPFF Record.***ParamValFuel > *Unit Capacity Record*.MaximumHourlyHeatInputCapacity
return result B
else
return result C

Results:

Result	Response	Severity
A	You did not report one and only one active Unit Capacity record in your monitoring plan for the unit (or for each unit linked to the pipe) for the hour.	Critical Error Level 1
B	Warning: The ParameterValueForFuel reported in the HPFF record for HI for FuelCode [fuelcd] exceeds the MaximumHourlyHeatInputCapacity reported in the Unit Capacity record in your monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of maximum values in the monitoring plan and make proper adjustments when necessary. You should investigate the cause of these exceedances and determine whether an adjustment to the MaximumHourlyHeatInputCapacity reported in your monitoring plan is necessary.	Informational Message
C	The ParameterValueForFuel reported in the HPFF record for HI for FuelCode [fuelcd] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
D	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
E	You reported a SourceOfDataVolumetricCode of 4 in the HFF record, indicating that you burned an emergency fuel. However, you did not report the maximum hourly heat input capacity for the unit as the ParameterValueforFuel in the HPFF record for HI, which is required when you burn an emergency fuel.	Critical Error Level 1
F	You reported a SourceOfDataMassCode of 4 in the HFF record, indicating that you burned an emergency fuel. However, you did not report the maximum hourly heat input capacity for the unit as the ParameterValueforFuel in the HPFF record for HI, which is required when you burn an emergency fuel.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-23

Check Name: Check Heat Input Units Of Measure

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current HI HPFF Record*** is not null)

If (***Current HI HPFF Record***.ParameterUOMCode <> "MMBTUHR")
return result A

Results:

Result	Response	Severity
A	The ParameterUOMCode reported in the HPFF record for HI for FuelCode [fuelcd] is missing or invalid. The value should be "MMBTUHR".	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-24

Check Name: Validate SO2 Record

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

Current SO2 HPFF Record = null
HFF SO2 Equation Code = null

Count the HourlyParamFuelFlow record where
HourlyParamFuelFlow.HourlyFuelFlowID = ***Current Fuel Flow Record***.HourlyFuelFlowID AND
HourlyParamFuelFlow.ParameterCode = "SO2"

If (*Count* > 1)
SO2 App D Accumulator = -1
return result A
Else If (*Count* == 0)
If (***SO2 App D Method Active For Hour*** == true)
SO2 App D Accumulator = -1
return result B
Else if (***SO2 App D Method Active For Hour*** == true)
Current SO2 HPFF Record = matching record
SO2 HPFF Exists = true

if (***Current SO2 HPFF Record***.MonitoringFormulaId is null
return result C
else
Cur SO2 Mon Formula Record = Lookup active formula in MonitoringFormula Table where
MonitoringFormulaID = ***Current SO2 HPFF Record***.MonitoringFormulaID

if (*Cur SO2 Mon Formula Record* is null)
return result D
else if (*Cur SO2 Mon Formula Record*.ParameterCode <> "SO2")
return result E
else if (***Current Fuel Group*** == "GAS")
If (*Cur SO2 Mon Formula Record*.EquationCode in set {D-4, D-5})
HFF SO2 Equation Code = *Cur SO2 Mon Formula Record*.EquationCode
else
return result F
else
If (*Cur SO2 Mon Formula Record*.EquationCode == "D-2")
HFF SO2 Equation Code = *Cur SO2 Mon Formula Record*.EquationCode
else
return result F
else
return result G

Results:

Result	Response	Severity
A	You reported more than one HPFF record for [parameter] for FuelCode [fuelcd] for the hour.	Critical Error Level 1
B	You did not report an HPFF record for [parameter] for FuelCode [fuelcd] for the hour.	Critical Error Level 1
C	You did not report a FormulaID in the HPFF record for SO2 for FuelCode [fuelcd].	Critical Error Level 1
D	You reported FormulaID [ID] in the HPFF record for SO2 for FuelCode [fuelcd], but there is no active Formula record for this formula in your monitoring plan.	Critical Error Level 1
E	You reported FormulaID [ID] in the HPFF record for SO2 for FuelCode [fuelcd], but this is not an SO2 formula.	Critical Error Level 1
F	The FormulaCode of FormulaID [ID] reported in the HPFF record for SO2 for FuelCode [fuelcd] is invalid.	Critical Error Level 1
G	You reported an HPFF record for SO2 for FuelCode [fuelcd], but you do not have an active Appendix D SO2 method for the hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-25

Check Name: Check Extraneous SO2 Record Fields

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

Hourly Extraneous Fields = null

If (***Current SO2 HPFF Record*** is not null)

If (***Current SO2 HPFF Record***.MonitoringSystemID is not null)
append "MonitoringSystemID" to ***Hourly Extraneous Fields***
If (***Current SO2 HPFF Record***.SegmentNumber is not null)
append "SegmentNumber" to ***Hourly Extraneous Fields***
If (***Current SO2 HPFF Record***.OperatingConditionCode is not null)
append "OperatingConditionCode" to ***Hourly Extraneous Fields***
If (***Current SO2 HPFF Record***.SampleTypeCode is not null)
append "SampleTypeCode" to ***Hourly Extraneous Fields***

If (***Hourly Extraneous Fields*** is not null)
return result A

Results:

Result	Response	Severity
A	You reported [fieldnames] in the HPFF record for SO2 for FuelCode [fuelcd]. This data should be blank.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-26

Check Name: Check SO2 Units Of Measure

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current SO2 HPFF Record*** is not null)

If (***Current SO2 HPFF Record***.ParameterUOMCode <> "LBHR")
return result A

Results:

Result	Response	Severity
A	The ParameterUOMCode reported in the HPFF record for SO2 for FuelCode [fuelcd] is missing or invalid. The value should be "LBHR".	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-27

Check Name: Calculate SO2 Mass Rate

Related Former Checks:

Applicability: Appendix D Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

HFF Calc SO2 = null

If (***Current SO2 HPFF Record*** is not null)
if (***Current Fuel Group*** == "GAS" AND ***HFF SO2 Equation Code*** == "D-4" AND ***HFF Sulfur*** is not null AND ***HFF Calc Volumetric Rate*** is not null

SO2 HPFF Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "SO2 Gas HPFF" AND
UOM = "LBHR"

HFF Calc SO2 = ***HFF Sulfur*** * ***HFF Calc Volumetric Rate*** * 2.0 / 7000, and round the result to 5 decimal places.

else if (***Current Fuel Group*** == "GAS" AND ***HFF SO2 Equation Code*** == "D-5" AND ***HFF SO2 Emission Rate*** is not null AND ***HFF Calc HI Rate*** is not null

SO2 HPFF Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "SO2 Gas HPFF" AND
UOM = "LBHR"

HFF Calc SO2 = ***HFF SO2 Emission Rate*** * ***HFF Calc HI Rate***, and round the result to 5 decimal places.

else if (***Current Fuel Group*** == "OIL" AND ***HFF Sulfur*** is not null AND ***HFF Calc Mass Oil Rate*** is not null

SO2 HPFF Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "SO2 Oil HPFF" AND
UOM = "LBHR"

HFF Calc SO2 = ***HFF Sulfur*** * ***HFF Calc Mass Oil Rate*** * 2.0 / 100, and round the result to 1 decimal place.

If (***HFF Calc SO2*** is not null)
If ***Current Fuel Flow Record***.FuelUsageTime > 0 AND ***Current Fuel Flow Record***.FuelUsageTime <= 1 AND ***SO2 App D Accumulator*** >= 0)
// Note - this accumulates totals for all Fuel flow records and does not work like a normal parameter
SO2 App D Accumulator = ***SO2 App D Accumulator*** + ***HFF Calc SO2*** * ***Current Fuel Flow Record***.FuelUsageTime
else
SO2 App D Accumulator = -1

If (***Current SO2 HPFF Record***.ParamValFuel >= 0)
if (ABS(***HFF Calc SO2*** - ***Current SO2 HPFF Record.***ParamValFuel) > *SO2 HPFF Tolerance*)
return result A
else
SO2 App D Accumulator = -1
return result B

Results:

Result	Response	Severity
A	The ParameterValueForFuel reported in the HPFF record for SO2 for FuelCode [fuelcd] is inconsistent with the recalculated value.	Critical Error Level 1
B	The ParameterValueForFuel in the HPFF record for [parameter] for FuelCode [fuelcd] could not be recalculated due to errors listed above.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-28

Check Name: Determine Sulfur Content

Related Former Checks:

Applicability: Appendix D Check

Description:

Validation Tables:

Fuel Type Reality Checks for Sulfur (Cross Check Table)
Fuel Type Warning Levels for Sulfur (Cross Check Table)
Table D-6 Missing Data Values (Cross Check Table)

Specifications:

HFF Sulfur = null
Current Sulfur Record = null

Count the HourlyParamFuelFlow record where
HourlyParamFuelFlow.HourlyFuelFlowID = ***Current Fuel Flow Record***.HourlyFuelFlowID AND
HourlyParamFuelFlow.ParameterCode = "SULFUR"

If (*Count* > 1)
return result A
Else If (*Count* == 0)
If (***HFF SO2 Equation Code*** in set {D-2, D-4})
return result B
Else if (***HFF SO2 Equation Code*** in set {D-2, D-4})

Current Sulfur Record = matching record
Sulfur UOM = ***Current Sulfur Record***.ParameterUOMCode

If (***Current Fuel Group*** == "GAS" AND *Sulfur UOM* <> "GRHSCF")
return result C
else if (***Current Fuel Group*** == "OIL" AND *Sulfur UOM* <> "PCT")
return result C
else if (***Current Sulfur Record***.ParamValFuel > 0)

Sulfur Default = null
If (*Sulfur UOM* == "GRHSCF")
Sulfur Precision = 1
else
Sulfur Precision = 4

If (***Current Sulfur Record***.SampleTypeCode == 8)
Sulfur Default = Lookup "MissingDataValue" in "Table D-6 Missing Data Values"
where "Parameter" column = "SULFUR" AND "FuelCode" column = ***Current Fuel Flow Record***.FuelCode

If (*Sulfur Default* == null)
Max Expected Sulfur = Lookup "Upper Value" in "Fuel Flow Warning Levels for Sulfur Content Cross Check Table"
where "Fuel Code" column = ***Current Fuel Flow Record***.FuelCode
Min Expected Sulfur = Lookup "Lower Value" in "Fuel Flow Warning Levels for Sulfur Content Cross Check Table"
where "Fuel Code" column = ***Current Fuel Flow Record***.FuelCode
Max Allowed Sulfur = Lookup "Upper Value" in "Fuel Flow Reality Checks for Sulfur Content Cross Check Table"
where "Fuel Code" column = ***Current Fuel Flow Record***.FuelCode

Min Allowed Sulfur = Lookup "Lower Value" in "Fuel Flow Reality Checks for Sulfur Content Cross Check Table"
where "Fuel Code" column = ***Current Fuel Flow Record***.FuelCode

If (*Max Allowed Sulfur* is not null AND ***Current Sulfur Record***.ParamValFuel > *Max Allowed Sulfur)* OR (*Min Allowed Sulfur* is not null AND ***Current Sulfur Record***.ParamValFuel < *Min Allowed Sulfur*)
return result D
else if (***Current Sulfur Record***.ParamValFuel is not rounded to *Sulfur Precision*)
return result I
else
HFF Sulfur = ***Current Sulfur Record***.ParamValFuel
if (*Min Expected Sulfur* is not null AND ***HFF Sulfur*** < *Min Expected Sulfur)* OR (*Max Expected Sulfur* is not null AND ***HFF Sulfur*** > *Max Expected Sulfur*)
return result E

else
If (***Current Sulfur Record***.ParamValFuel is not rounded to *Sulfur Precision*)
return result I
else if *Sulfur Default* == ***Current Sulfur Record***.ParamValFuel
HFF Sulfur == ***Current Sulfur Record***.ParamValFuel
else
return result F
else
return result G
else
return result H

Results:

Result	Response	Severity
A	You reported more than one HPFF record for [parameter] for FuelCode [fuelcd] for the hour.	Critical Error Level 1
B	You reported a formula with a FormulaCode of [code] in the HPFF record for SO2 for FuelCode [fuelcd], but you did not report an HPFF record for SULFUR. Use of this formula to calculate SO2 requires the reporting of the fuel's sulfur content.	Critical Error Level 1
C	The ParameterUOMCode reported in the HPFF record for SULFUR for FuelCode [fuelcd] is missing or invalid.	Critical Error Level 1
D	The ParameterValueForFuel reported in the HPFF record for [parameter] for FuelCode [fuelcd] is outside the range of allowable values for the fuel type.	Critical Error Level 1
E	The ParameterValueForFuel reported in the HPFF record for [parameter] for FuelCode [fuelcd] is outside the range of expected values for the fuel type.	Critical Error Level 2
F	You reported a SampleTypeCode of 8 in the HPFF record for [parameter] for FuelCode [fuelcd], indicating the use of a Table D-6 default, but the ParameterValueForFuel does not equal the default value for the fuel.	Critical Error Level 1
G	The ParameterValueForFuel reported in the HPFF record for [parameter] for FuelCode [fuelcd] is invalid. The value should be greater than 0.	Critical Error Level 1
H	You reported an HPFF record for [parameter] for FuelCode [fuelcd], but you do not require this value to calculate SO2.	Critical Error Level 1
I	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-29

Check Name: Check Extraneous Sulfur Record Fields

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

Hourly Extraneous Fields = null

If (***Current Sulfur Record*** is not null)

    If (***Current Sulfur Record***.FormulaIdentifier is not null)
        append "FormulaIdentifier" to ***Hourly Extraneous Fields***
    If (***Current Sulfur Record***.MonitoringSystemID is not null)
        append "MonitoringSystemID" to ***Hourly Extraneous Fields***
    If (***Current Sulfur Record***.SegmentNumber is not null)
        append "SegmentNumber" to ***Hourly Extraneous Fields***
    If (***Current Sulfur Record***.OperatingConditionCode is not null)
        append "OperatingConditionCode" to ***Hourly Extraneous Fields***

    If (***Hourly Extraneous Fields*** is not null)
        return result A

Results:

Result	Response	Severity
A	You reported [fieldnames] in the HPFF record for SULFUR for FuelCode [fuelcd]. This data should be blank.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-30

Check Name: Check Sulfur Sample Type

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current Sulfur Record*** is not null)
    If (***Current Fuel Group*** == "OIL" AND ***Current Sulfur Record***.SampleTypeCode not in {1, 2, 5, 6, 7, 8})
        return result A
    else if (***Current Fuel Group*** == "GAS" AND ***Current Sulfur Record***.SampleTypeCode not in {0, 2, 4, 5, 6, 7, 8})
        return result A

Results:

Result	Response	Severity
A	The SampleTypeCode reported in the HPFF record for SULFUR for FuelCode [fuelcd] is missing or invalid.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-31

Check Name: Determine SO2 Emission Rate

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

HFF SO2 Emission Rate = null
Current SO2R Record = null

Count the HourlyParamFuelFlow record where
HourlyParamFuelFlow.HourlyFuelFlowID = ***Current Fuel Flow Record***.HourlyFuelFlowID AND
HourlyParamFuelFlow.ParameterCode = "SO2R"

If (*Count* > 1)
return result A
Else If (*Count* == 0)
If (***HFF SO2 Equation Code*** == "D-5")
return result B
Else if (***HFF SO2 Equation Code*** == "D-5")

Current SO2R Record = matching record
If (***Current SO2R Record***.ParameterUOMCode <> "LBMMBTU"}
return result C
else if (***Current SO2R Record***.ParamValFuel > 0)
HFF SO2 Emission Rate = ***Current SO2R Record***.ParamValFuel
else
return result D
else
return result E

Results:

Result	Response	Severity
A	You reported more than one HPFF record for [parameter] for FuelCode [fuelcd] for the hour.	Critical Error Level 1
B	You reported a formula with a FormulaCode of "D-5" in the HPFF record for SO2 for FuelCode [fuelcd], but you did not report an HPFF record for SO2R. Use of formula D-5 to calculate SO2 requires the reporting of the SO2 emission rate for the fuel.	Critical Error Level 1
C	The ParameterUOMCode reported in the HPFF record for SO2R for FuelCode [fuelcd] is missing or invalid. The value should be "LBMMBTU".	Critical Error Level 1
D	The ParameterValueForFuel reported in the HPFF record for [parameter] for FuelCode [fuelcd] is invalid. The value should be greater than 0.	Critical Error Level 1
E	You reported an HPFF record for [parameter] for FuelCode [fuelcd], but you do not require this value to calculate SO2.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-32

Check Name: Check Extraneous SO2R Record Fields

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

Hourly Extraneous Fields = null

If (***Current SO2R Record*** is not null)

If (***Current SO2R Record***.MonitoringSystemID is not null)
append "MonitoringSystemID" to ***Hourly Extraneous Fields***
If (***Current SO2R Record***.SegmentNumber is not null)
append "SegmentNumber" to ***Hourly Extraneous Fields***
If (***Current SO2R Record***.OperatingConditionCode is not null)
append "OperatingConditionCode" to ***Hourly Extraneous Fields***
If (***Current SO2R Record***.SampleTypeCode is not null)
append "SampleTypeCode" to ***Hourly Extraneous Fields***

If (***Hourly Extraneous Fields*** is not null)
return result A

Results:

Result	Response	Severity
A	You reported [fieldnames] in the HPFF record for SO2R for FuelCode [fuelcd]. This data should be blank.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-33

Check Name: Check SO2R Formula

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current SO2R Record*** is not null)
    If (***Current SO2R Record***.FormulaIdentifier is null)
        If (***Current Fuel Flow Record***.FuelCode <> "PNG" OR ***Current SO2R Record***.ParamValFuel <> 0.0006)
            return result A
    else

        Cur SO2R Mon Formula Record = Lookup active formula in MonitoringFormula Table where
            MonitoringFormulaID = ***Current SO2R Record***.MonitoringFormulaID

        if (*Cur SO2R Mon Formula Record* is null)
            return result B
        else if (*Cur SO2R Mon Formula Record*.ParameterCode <> "SO2R"
            return result C
        else if (*Cur SO2R Mon Formula Record*.EquationCode <> "D-1H")
            return result D

Results:

Result	Response	Severity
A	You did not report a FormulaID in the HPFF record for SO2R for FuelCode [fuelcd]. This formula is required except when using the standard default emission rate of 0.0006 for pipeline natural gas.	Critical Error Level 1
B	You reported FormulaID [ID] in the HPFF record for SO2R for FuelCode [fuelcd], but there is no active Formula record for this formula in your monitoring plan.	Critical Error Level 1
C	You reported FormulaID [ID] in the HPFF record for SO2R for FuelCode [fuelcd], but this is not an SO2R formula.	Critical Error Level 1
D	The FormulaCode of FormulaID [ID] reported in the HPFF record for SO2R for FuelCode [fuelcd] is invalid. The FormulaCode should be "D-1H".	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-34

Check Name: Check Reported SO2 Mass Rate

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current SO2 HPFF Record*** is not null)

If (***Current SO2 HPFF Record***.ParamValFuel is null or is less than 0
return result A
else if (***Current Fuel Group*** == "OIL" AND ***Current SO2 HPFF Record***.ParamValFuel is not rounded to one decimal place)
return result B

Results:

Result	Response	Severity
A	The ParameterValueForFuel reported in the HPFF record for SO2 for FuelCode [fuelcd] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
B	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-35

Check Name: Determine FC Factor

Related Former Checks:

Applicability: Appendix D Check

Description:

Validation Tables:

Fuel Type Reality Checks for FC FACTOR (Cross Check Table)

Specifications:

HFF Fc Factor = null
Current Fc Factor Record = null

Count the HourlyParamFuelFlow record where
HourlyParamFuelFlow.HourlyFuelFlowID = ***Current Fuel Flow Record***.HourlyFuelFlowID AND
HourlyParamFuelFlow.ParameterCode = "FC"

If (*Count* > 1)
return result A
Else If (*Count* == 0)
If (***Current CO2 HPFF Record*** is not null)
return result B
Else if (***Current CO2 HPFF Record*** is not null)

Current Fc Factor Record = matching record

If (***Current Fc Factor Record***.ParameterUOMCode <> "SCFCBTU")
return result C
else if (***Current FcFactor Record***.ParamValFuel > 0)

if (***Current FcFactor Record***.ParamValFuel is not rounded to one decimal place)
return result G

else
HFF Fc Factor = ***Current Fc Factor Record***.ParamValFuel

Max Allowed Fc Factor = Lookup "Upper Value" in "Fuel Type Reality Checks for FC Factor Cross Check Table"
where "FuelType" column = ***Current Fuel Group***
Min Allowed Fc Factor = Lookup "Lower Value" in "Fuel Type Reality Checks for FC Factor Cross Check Table"
where "FuelType" column = ***Current Fuel Group***

If (*Max Allowed FcFactor* is not null AND ***Current Fc Factor Record***.ParamValFuel > *Max Allowed Fc Factor)*
OR (*Min Allowed FcFactor* is not null AND ***Current Fc Factor Record***.ParamValFuel < *Min Allowed Fc Factor*)
return result D
else
return result E
else
return result F

Results:

Result	Response	Severity
A	You reported more than one HPFF record for [parameter] for FuelCode [fuelcd] for the hour.	Critical Error Level 1
B	You reported an HPFF record for CO2 for FuelCode [fuelcd], but you did not report an HPFF record for FC for the hour.	Critical Error Level 1
C	The ParameterUOMCode reported in the HPFF record for FC for FuelCode [fuelcd] is missing or invalid.	Critical Error Level 1
D	The ParameterValueForFuel reported in the HPFF record for [parameter] for FuelCode [fuelcd] is outside the range of allowable values for the fuel type.	Critical Error Level 1
E	The ParameterValueForFuel reported in the HPFF record for FC for FuelCode [fuelcd] is invalid.	Critical Error Level 1
F	You reported an HPFF record for FC for FuelCode [fuelcd], but you have not reported an HPFF record for CO2 for the hour.	Critical Error Level 1
G	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-36

Check Name: Check Extraneous Fc Factor Record Fields

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

Hourly Extraneous Fields = null

If (***Current Fc Factor Record*** is not null)

If (***Current Fc Factor Record***.FormulaIdentifier is not null)
append "FormulaIdentifier" to ***Hourly Extraneous Fields***
If (***Current Fc Factor Record***.MonitoringSystemID is not null)
append "MonitoringSystemID" to ***Hourly Extraneous Fields***
If (***Current Fc Factor Record***.SegmentNumber is not null)
append "SegmentNumber" to ***Hourly Extraneous Fields***
If (***Current Fc Factor Record***.OperatingConditionCode is not null)
append "OperatingConditionCode" to ***Hourly Extraneous Fields***
If (***Current Fc Factor Record***.SampleTypeCode is not null)
append "SampleTypeCode" to ***Hourly Extraneous Fields***

If (***Hourly Extraneous Fields*** is not null)
return result A

Results:

Result	Response	Severity
A	You reported [fieldnames] in the HPFF record for FC for FuelCode [fuelcd]. This data should be blank.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-37

Check Name: Validate CO2 Record

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

Current CO2 HPFF Record = null

Count the HourlyParamFuelFlow record where
HourlyParamFuelFlow.HourlyFuelFlowID = ***Current Fuel Flow Record***.HourlyFuelFlowID AND
HourlyParamFuelFlow.ParameterCode = "CO2"

If (*Count* > 1)
CO2 App D Accumulator = -1
return result A
Else If (*Count* == 0)
If (***CO2 App D Method Active For Hour*** == true)
CO2 App D Accumulator = -1
If (***Legacy Data Evaluation*** == false)
return result B
else
return result H
Else if (***CO2 App D Method Active For Hour*** == true)
Current CO2 HPFF Record = matching record
CO2 HPFF Exists = true

if (***Current CO2 HPFF Record***.MonitoringFormulaId is null
return result C
else
Cur CO2 Mon Formula Record = Lookup active formula in MonitoringFormula Table where
MonitoringFormulaID = ***Current CO2 HPFF Record***.MonitoringFormulaID

if (*Cur CO2 Mon Formula Record* is null)
return result D
else if (*Cur CO2 Mon Formula Record*.ParameterCode <> "CO2")
return result E
else if (*Cur CO2 Mon Formula Record*.EquationCode <> "G-4")
return result F
else
return result G

Results:

Result	Response	Severity
A	You reported more than one HPFF record for [parameter] for FuelCode [fuelcd] for the hour.	Critical Error Level 1
B	Your monitoring plan indicates the use of the Appendix D CO2 method, but you did not report an HPFF record for CO2 for FuelCode [fuelcd] for the hour.	Critical Error Level 1
C	You did not report a FormulaID in the HPFF record for CO2 for FuelCode [fuelcd].	Critical Error Level 1
D	You reported FormulaID [ID] in the HPFF record for CO2 for FuelCode [fuelcd], but there is no active Formula record for this formula in your monitoring plan.	Critical Error Level 1
E	You reported FormulaID [ID] in the HPFF record for CO2 for FuelCode [fuelcd], but this is not a CO2 formula.	Critical Error Level 1
F	The FormulaCode of FormulaID [ID] reported in the HPFF record for CO2 for FuelCode [fuelcd] is invalid. The FormulaCode should be "G-4".	Critical Error Level 1
G	You reported an HPFF record for CO2 for FuelCode [fuelcd], but you do not have an active Appendix D CO2 method for the hour.	Critical Error Level 1
H	Your monitoring plan indicates the use of the Appendix D CO2 method, but you did not report an HPFF record for CO2 for FuelCode [fuelcd] for the hour. Fuel-specific CO2 emissions data was not required in the EDR data, but is required for all data submitted through ECMPS. The software will not recalculate CO2 emissions values.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-38

Check Name: Check Extraneous CO2 Record Fields

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

Hourly Extraneous Fields = null

If (***Current CO2 HPFF Record*** is not null)

If (***Current CO2 HPFF Record***.MonitoringSystemID is not null)
append "MonitoringSystemID" to ***Hourly Extraneous Fields***
If (***Current CO2 HPFF Record***.SegmentNumber is not null)
append "SegmentNumber" to ***Hourly Extraneous Fields***
If (***Current CO2 HPFF Record***.OperatingConditionCode is not null)
append "OperatingConditionCode" to ***Hourly Extraneous Fields***
If (***Current CO2 HPFF Record***.SampleTypeCode is not null)
append "SampleTypeCode" to ***Hourly Extraneous Fields***

If (***Hourly Extraneous Fields*** is not null)
return result A

Results:

Result	Response	Severity
A	You reported [fieldnames] in the HPFF record for CO2 for FuelCode [fuelcd]. This data should be blank.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-39

Check Name: Calculate CO2 Mass Rate

Related Former Checks:

Applicability: Appendix D Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

HFF Calc CO2 = null

If (***Current CO2 HPFF Record*** is not null)
if (***HFF Calc HI Rate*** is not null AND ***HFF Fc Factor*** is not null)

CO2 HPFF Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "CO2" AND
UOM = "TNHR"

***HFF Calc CO2 = HFF Calc HI Rate * HFF Fc Factor** * 44.0 / (385.0 * 2000.0), and round the result to one decimal place.

If ***Current Fuel Flow Record***.FuelUsageTime > 0 AND ***Current Fuel Flow Record***.FuelUsageTime <= 1 AND ***CO2 App D Accumulator*** >= 0)
// Note - this accumulates totals for all Fuel flow records and does not work like a normal parameter
CO2 App D Accumulator = ***CO2 App D Accumulator*** + ***HFF Calc CO2 * Current Fuel Flow Record***.FuelUsageTime
else
CO2 App D Accumulator = -1

If (***Current CO2 HPFF Record***.ParamValFuel >= 0)
if (ABS(***HFF Calc CO2 - Current CO2 HPFF Record.***ParamValFuel) > *CO2 HPFF Tolerance*)
return result A
else
CO2 App D Accumulator = -1
return result B

Results:

Result	Response	Severity
A	The ParameterValueForFuel reported for HPFF record for CO2 for FuelCode [fuelcd] is inconsistent with the recalculated value.	Critical Error Level 1
B	The ParameterValueForFuel in the HPFF record for [parameter] for FuelCode [fuelcd] could not be recalculated due to errors listed above.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-40

Check Name: Check Reported CO2 Mass Rate

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current CO2 HPFF Record*** is not null)

If (***Current CO2 HPFF Record***.ParamValFuel is null or is less than 0
return result A
else if (***Current CO2 HPFF Record***.ParamValFuel is not rounded to one decimal place)
return result B

Results:

Result	Response	Severity
A	The ParameterValueForFuel reported in the HPFF record for CO2 for FuelCode [fuelcd] is invalid. The value should be greater than or equal to 0.	Critical Error Level 1
B	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-44

Check Name: Check CO2 Units Of Measure

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current CO2 HPFF Record*** is not null)

If (***Current CO2 HPFF Record***.ParameterUOMCode <> "TNHR")
return result A

Results:

Result	Response	Severity
A	The ParameterUOMCode reported in the HPFF record for CO2 for FuelCode [fuelcd] is missing or invalid. The value should be "TNHR".	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-45

Check Name: Determine Appendix D Measure Codes

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current Fuel Flow Record***.SourceOfDataMassCode in set {4, 5, 6} OR ***Current Fuel Flow Record***.SourceOfDataVolumetricCode in set {4, 5, 6} OR ***Monitor Measure Code Array*** for "FF" =="OTHER")
    set ***Monitor Measure Code Array*** for "FF" to "OTHER"
else if (***Current Fuel Flow Record***.SourceOfDataMassCode in set {1, 3} OR ***Current Fuel Flow Record***.SourceOfDataVolumetricCode in set {1, 3})
    if (***Monitor Measure Code Array*** for "FF" begins with "MEAS")
        set ***Monitor Measure Code Array*** for "FF" to "MEASSUB"
    else
        set ***Monitor Measure Code Array*** for "FF" to "SUB"
else if (***Current Fuel Flow Record***.SourceOfDataMassCode in set {0, 9} OR ***Current Fuel Flow Record***.SourceOfDataVolumetricCode in set {0, 9}))
    if (***Monitor Measure Code Array*** for "FF" contains "SUB")
        set ***Monitor Measure Code Array*** for "FF" to "MEASSUB"
    else
        set ***Monitor Measure Code Array*** for "FF" to "MEASURE"

if (***Current Sulfur Record*** is not null)
    if (***Current Sulfur Record***.SampleTypeCode == 8)
        if (***Monitor Measure Code Array*** for "SULFUR" begins with "MEAS")
            set ***Monitor Measure Code Array*** for "SULFUR" to "MEASSUB"
        else
            set ***Monitor Measure Code Array*** for "SULFUR" to "SUB"
    else if (***Current Sulfur Record***.SampleTypeCode in set {0, 1, 2, 4, 5, 6, 7})
        if (***Monitor Measure Code Array*** for "SULFUR" contains "SUB")
            set ***Monitor Measure Code Array*** for "SULFUR" to "MEASSUB"
        else
            set ***Monitor Measure Code Array*** for "SULFUR" to "MEASURE"

if (***Current GC V Record*** is not null)
    if (***Current GC V Record***.SampleTypeCode == 8)
        if (***Monitor Measure Code Array*** for "GCV" begins with "MEAS")
            set ***Monitor Measure Code Array*** for "GCV" to "MEASSUB"
        else
            set ***Monitor Measure Code Array*** for "GCV" to "SUB"
    else if (***Current GCV Record***.SampleTypeCode in set {0, 1, 2, 3, 4, 5, 6, 7})
        if (***Monitor Measure Code Array*** for "GCV" contains "SUB")
            set ***Monitor Measure Code Array*** for "GCV" to "MEASSUB"
        else
            set ***Monitor Measure Code Array*** for "GCV" to "MEASURE"

if (***Current Density Record*** is not null)
    if (***Current Density Record***.SampleTypeCode == 8)
        if (***Monitor Measure Code Array*** for "DENSITY" begins with "MEAS")
            set ***Monitor Measure Code Array*** for "DENSITY" to "MEASSUB"
        else
            set ***Monitor Measure Code Array*** for "DENSITY" to "SUB"
    else if (***Current Density Record***.SampleTypeCode in set {1, 2, 5, 6, 7})
        if (***Monitor Measure Code Array*** for "DENSITY" contains "SUB")

set ***Monitor Measure Code Array*** for "DENSITY" to "MEASSUB"

else

set ***Monitor Measure Code Array*** for "DENSITY" to "MEASURE"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAD-46

Check Name: Update System Supplemental Data for Hourly Fuel Flow

Related Former Checks:

Applicability: Appendix D Check

Description: Updates the System Operarting Supplemental Data for the system reported with the Current Hourly Fuel Flow reocrd.

Specifications:

If ***DerivedHourlyChecksNeeded*** is true AND ***CurrentOperatingTime*** is greater than 0,

If ***CurrentFuelFlowRecord***.MonitoringSystemID is not null,

Set *SupplementalDataDictionary* to the dictionary at ***CurrentMonitorPlanLocationPosition*** in ***SystemOperatingSuppDataDictionaryArray***.

If *SupplementalDataDictionary* contains key ***CurrentFuelFlowRecord***.MonitoringSystemID,

Set *SupplementalDataRecord* in *SupplementalDataDictionary* value where key is equal to ***CurrentFuelFlowRecord***.MonitoringSystemID.

Else

Create a new *SupplementalDataRecord* with MonitoringSystemID equal to ***CurrentFuelFlowRecord***.MonitoringSystemID, and OpDays, OpHours, OsDays and OsHours equal to 0..
Add *SupplementalDataRecord* to *SupplementalDataDictionary* with a key of ***CurrentFuelFlowRecord***.MonitoringSystemID.

Increament *SupplementalDataRecord*.QuarterlyOperatingCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.QuarterlyOperatingCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.MayAndJuneOperatingCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.

Increament *SupplementalDataRecord*.MayAndJuneOperatingCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.

Results:

Result	Response	Severity

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Category:

Hourly Appendix E

Check Code: HOURAE-1

Check Name: Initialize AE Reporting Method

Related Former Checks:

Applicability: Appendix E Check

Description: Determines whether Appendix E Reporting is from a single fuel source, multiple fuel sources, or a Constant Mix Fuel Source

Specifications:

App E Reporting Method = null
App E Op Code = null
App E Segment Number = null
App E Reported Value = null
App E Fuel Code = null
App E Calc HI = null

if (***Current NOx Rate Method Code*** == "AE")

Total Fuel Sources = ***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil***

// App E Constant Fuel Mix detected when processing DHV records
if (*Total Fuel Sources* > 1)
App E Reporting Method = "MULTIPLE"
else if (*Total Fuel Sources* == 1)
App E Reporting Method = "SINGLE"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOURAE-2

Check Name: Validate NOXR Record

Related Former Checks:

Applicability: Appendix E Check

Description: Locates the appropriate NOXR HourlyParamFuelFlow record for current fuel flow record

Specifications:

Current App E NOXR Record = null

App E NOXR HPFF Count for Gas = find matching HourlyParamFuelFlow records where
HourlyParamFuelFlow.HourlyFuelFlowID = ***Current Fuel Flow Record***.HourlyFuelFlowID AND
HourlyParamFuelFlow.ParameterCode = "NOXR"

If (*App E NOXR HPFF Count for Gas* == 0)
if (***App E Reporting Method*** in set {MULTIPLE, SINGLE})
NOXR App E Accumulator = -1
return result A

else if (If *App E NOXR HPFF Count for Gas* > 1)
if (***App E Reporting Method*** in set {MULTIPLE, SINGLE})
NOXR App E Accumulator = -1
return result B
else
return result D

Else if (***App E Reporting Method*** in set {MULTIPLE, SINGLE})

Current App E NOXR Record = matching record
App E Segment Number = ***Current App E NOXR Record.***SegmentNumber
App E Reported Value = Current App E NOXR Record.ParamValFuel
App E Calc HI = ***HFF Calc HI Rate***
App E Fuel Code = Current Fuel Flow Record.FuelCode

if ***Current Appe E NOXR Record.***OperatingConditionCode in set {E, X, Y, Z, U, W, N, M}
App E Op Code = ***Current App E NOXR Record.***OperatingConditionCode
else
App E Op Code = null
return result C

else if (***App E Reporting Method*** == "CONSTANT")
return result D
else
return result E

Results:

Result	Response	Severity
A	You did not report an HPFF record for NOXR to report the NOx emission rate for individual fuels.	Critical Error Level 1
B	You reported more than one HPFF record for NOXR for FuelCode [fuelcd] for the hour.	Critical Error Level 1
C	The OperatingConditionCode reported in the HPFF record for NOXR for FuelCode [FUELCD] is missing or invalid.	Critical Error Level 1
D	You reported an HPFF record for NOXR, but, according to your monitoring plan, you use an Appendix E mixed fuel curve to determine the NOx emission rate. If this is the case, you should report the NOx emission rate in a NOXR DHV record. The HPFF record will not be evaluated and the NOx emissions rate will not be recalculated.	Critical Error Level 1
E	You reported an HPFF record for NOXR, but you have not defined a NOXR AE method in your monitoring plan that is active during the current hour. The HPFF record will not be evaluated and the NOx emissions rate will not be recalculated.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAE-4

Check Name: Check for Extraneous Fields in NOXR Record

Related Former Checks:

Applicability: Appendix E Check

Description:

Specifications:

Hourly Extraneous Fields = null

if (***Current App E NOXR Record*** is not null)
    if (***Current App E NOXR Record***.SampleTypeCode is not NULL)
        append "SampleTypeCode" to ***Hourly Extraneous Fields***

    if (***Current App E NOXR Record***.MonitoringFormulaId is not NULL)
        append "MonitoringFormulaID" to ***Hourly Extraneous Fields***

    If (***Hourly Extraneous Fields is not null***)
        return result A

Results:

Result	Response	Severity
A	You reported [fieldnames] in the HPFF record for NOXR for FuelCode [fuelcd]. This data should be blank.	Non-Critical Error

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAE-5

Check Name: Check Monitoring System Data for Appendix E NOXR

Related Former Checks:

Applicability: Appendix E Check

Description: Verifies whether monitoring system reported for Appendix E NOXR meets the reporting requirements

Specifications:

```
if (Current App E NOXR Record is not null)

    App E NOXE System ID = null
    App E NOXE System Identifier = null

    if (Current App E NOXR Record.MonitoringSystemId is null)
        if (Current App E NOXR Record.OperatingConditionCode == "E")
            if (HFF Fuel Indicator Code <> "E")
                return result A
        else
            return result B
    // report Monitoring System in all other cases
    else
        Current App E NOXR Mon Sys Record = find MonitoringSystem record where
            MonitoringSystem.MonitoringSystemId = Current App E NOXR Record .MonitoringSystemId
        if (Current App E NOXR Mon Sys Record is null)
            return result C
        else if (Current App E NOXR Mon Sys Record.SystemTypeCode <> "NOXE"
            return result D
        else if (Current App E NOXR Mon Sys Record.FuelCode <> Current Fuel Flow Record.FuelCode
            return result E
        else
            App E NOXE System ID = Current App E NOXR Record .MonitoringSystemId
            App E NOXE System Identifier = Current App E NOXR Mon Sys Record.SystemIdentifier
```

Results:

Result	Response	Severity
A	The OperatingConditionCode of E reported in the HPFF record for NOXR for FuelCode [fuelcd] indicates that the fuel is an emergency fuel, but this is inconsistent with the IndicatorCode in the UnitFuel record for the fuel.	Critical Error Level 1
B	You did not report a MonitoringSystemID in the HPFF record for NOXR for FuelCode [fuelcd], but you did not report an OperatingConditionCode of E. You must report a NOXE MonitoringSystemID for non-emergency fuels.	Critical Error Level 1
C	You reported MonitoringSystemID [ID] in the HPFF record for NOXR for FuelCode [fuelcd], but there is no MonitorSystem record for this system in your monitoring plan that was active during the hour.	Critical Error Level 1
D	You reported MonitoringSystemID [ID] in the HPFF record for NOXR for FuelCode [fuelcd], but this system is not a NOXE monitoring system.	Critical Error Level 1
E	You reported NOXE MonitoringSystemID [ID] in the HPFF record for NOXR, but the FuelCode of this system is not equal to FuelCode [fuelcd] in the associated HFF record.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAE-7

Check Name: Retrieve Appendix E Correlation Test Results or Default Value

Related Former Checks:

Applicability: Appendix E Check

Description: Finds most recent successful test results for Appendix E Tests in QASupplementalData or TestSummary Records

Specifications:

Maximum App E Curve NOx Emission Rate = null
App E NOx MER = null
App E Segment Total = null

If (***App E Op Code*** in set {N, W, X, Y, Z})

if (***Current Appendix E Status*** begins with "IC" or "Undetermined")

QA Supp Attribute Count Record = matching record in QASuppAttribute where
QASuppAttribute.QASuppDataId = ***Prior Appendix E Record***.QASuppDataId
QASuppAttribute.AttributeName = "SEGMENT_COUNT"

if (*QA Supp Attribute Count Record* is not null)

App E Segment Total = *QA Supp AttributeCount Record*.AttributeValue
Dimension ***App E Correlation NOx Rate Array*** with ***App E Segment Total*** elements
Dimension ***App E Correlation Heat Input Array*** with ***App E Segment Total*** elements

for (X = 1 to ***App E Segment Total***)
QA Supp Attribute Segment NOx Record = matching record in QASuppAttribute where
QASuppAttribute.QASuppDataId = ***Prior Appendix E Record***.QASuppDataId
QASuppAttribute.AttributeName = "NOX_RATE_X" (where X matches the loop variable)

if (*QA Supp Attribute Segment NOx Record* is not null)
if (*QA Supp Attribute Segment NOx Record*.AttributeValue >***Maximum App E Curve NOx Emission Rate***
Maximum App E Curve NOx Emission Rate = *QA Supp Attribute Segment NOx Record*.AttributeValue

App E Correlation NOx Rate Array[X] = *QA Supp Attribute Segment NOx Record*.AttributeValue

QA Supp Attribute Segment HI Record = matching record in QASuppAttribute where
QASuppAttribute.QASuppDataId = ***Prior Appendix E Record***.QASuppDataId
QASuppAttribute.AttributeName = "HI_RATE_X" (where X matches the loop variable)

if (*QA Supp Attribute Segment HI Record* is not null)
App E Correlation Heat Input Array[X] = *QA Supp Attribute Segment HI Record*.AttributeValue

else if (***App E Op Code*** in set {E, M, U})

Count active MonitoringDefault record for location where
ParameterCode = "NORX" AND
DefaultPurpose = "MD" AND
FuelCode = ***App E Fuel Code***

if (*Count* <> 1)

return result A
else
App E NOx MER Default Record = matching record
If (*App E NOx MER Default Record*.DefaultValue > 0
App E NOx MER = *App E NOx MER Default Record*.DefaultValue
else
return result B

Results:

Result	Response	Severity
A	The NOx emission rate could not be determined, because you did not report one and only one missing data default record for NORX for FuelCode [fuelcd] in your monitoring plan that was active during current hour.	Critical Error Level 1
B	The NOx emission rate could not be determined, because the DefaultValue in the NORX default record for FuelCode [fuelcd] is invalid.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation
	Conditions:	App E Checks Needed Equals true
2	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification
	Conditions:	App E Constant Fuel Mix Equals true
3	Process/Category:	Emissions Data Evaluation Report --------- Hourly Fuel Flow
	Conditions:	App E Constant Fuel Mix Equals false

Check Code: HOURAE-8

Check Name: Determine Appendix E Curve Segment

Related Former Checks:

Applicability: Appendix E Check

Description:

Specifications:

App E Calc Segment Num = null

if (***App E Op Code*** is not null)

switch (***App E Op Code***)
case "E" or "U" or "M" or "W":

if ***App E Segment Number*** is not null
return result A

case "N" or "X":

If (***App E Segment Total*** is not null)

if ***App E Segment Number*** is not null

if [***App E Segment Number*** > number of elements in the ***App E Correlation NOx Rate Array***)
return result B

else if ***(App E Correlation NOx Rate Array*** [***App E Segment Number***] <> ***Maximum App E Curve NOx Emission Rate***)
return result B

else
if (***Legacy Data Evaluation*** == false)
return result G

case "Y" OR "Z":

If (***App E Calc HI*** is not null) and (***App E Segment Total*** is not null)

i = 1
while (*i* <= ***App E Segment Total*** AND ***App E C alc HI*** > ***App E Correlation Heat Input Rate Array***[*i*])
i = *i* + 1

if (*i* <= ***App E Segment Total*** AND ***App E Calc HI*** <= ***App E Correlation Heat Input Array***[*i*])
App E Calc Segment Num = *i*

if (***App E Op Code*** == "Z")
if (***App E Calc Segment Num*** <> 1)
return result C
else if ***App E Segment Number*** is null
if (***Legacy Data Evaluation*** == false)
return result G
else if (***App E Segment Number*** <> 1)
return result D

else if ***App E Segment Number*** is null

```
                                    if (Legacy Data Evaluation == false)
                                        return result G

                                else if (App E Calc HI == App E Correlation Heat Input Array[i])
                                    if (App E Segment Number <> App E Calc Segment Num AND App E Segment
                                    Number <> App E Calc Segment Num + 1)
                                        return result E
                                else
                                    if (App E Segment Number <> App E Calc Segment Num)
                                        return result E

                            else
                                return result F
```

Results:

Result	Response	Severity
A	You reported a SegmentNumber in the HPFF or DHV record for NOXR for FuelCode [fuelcd]. This field should be blank when OperatingConditionCode is [OpCode].	Critical Error Level 1
B	You reported an OperatingConditionCode of [OpCode] in the DHV or HPFF record for NOXR for FuelCode [fuelcd], but the reported SegmentNumber does not represent the segment on the Appendix E curve with the maximum NOx emission rate.	Non-Critical Error
C	You reported an OperatingConditionCode of Z in the DHV or HPFF record for NOXR for FuelCode [fuelcd], but the calculated heat input rate is not below the lowest point on the Appendix E curve.	Critical Error Level 1
D	You reported an OperatingConditionCode of Z in the DHV or HPFF record for NOXR for FuelCode [fuelcd], but you did not report a SegmentNumber of 1.	Critical Error Level 1
E	The SegmentNumber reported in the HPFF or DHV record for NOXR for FuelCode [fuelcd] is inconsistent with the calculated heat input.	Critical Error Level 1
F	You reported an OperatingConditionCode of [OpCode] in the DHV or HPFF record for NOXR for FuelCode [fuelcd], but the calculated heat input rate is higher than the maximum heat input rate on the Appendix E curve. You should report an OperatingConditionCode of W, and use the appropriate substitute data algorithm to calculate the NOx emission rate.	Critical Error Level 1
G	You did not report a SegmentNumber in the HPFF or DHV record for NOXR for FuelCode [fuelcd].	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation
	Conditions:	App E Checks Needed Equals true
2	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification
	Conditions:	App E Constant Fuel Mix Equals true
3	Process/Category:	Emissions Data Evaluation Report --------- Hourly Fuel Flow
	Conditions:	App E Constant Fuel Mix Equals false

Check Code: HOURAE-9

Check Name: Calculate Appendix E NOx Rate

Related Former Checks:

Applicability: Appendix E Check

Description: For Appendix E data that was extrapolated to the piecewise linear curve under conditions where the Operating Condition Code was reported as ""Y", this check ensures that the reported value matches the calculated value based on heat input.

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

App E Calculated NOx Rate for Source = null

if (***App E Op Code*** in set {Y, Z})

If (***App E Calc Segment Num*** is not null)

if (***App E Calc Segment Num*** == 1)
App E Calculated NOx Rate for Source = App E Correlation NOx Rate Array[1]
else

y2 = ***App E Correlation NOx Rate Array***[***App E Calc Segment Num***]
x2 = ***App E Correlation Heat Input Array***[***App E Calc Segment Num***]
y1 = ***App E Correlation NOx Rate Array***[***App E Calc Segment Num*** - 1]
x1 = ***App E Correlation Heat Input Array***[***App E Calc Segment Num*** - 1]

slope = (y2 - y1) / (x2 - x1)

App E Calculated NOx Rate for Source = *slope* * (***App E Calc HI*** - x1) + y1, and round the result to 3 decimal places.

else if (***App E Op Code*** in set {N, X})

App E Calculated NOx Rate for Source = ***Maximum App E Curve NOx Emission Rate***

else if (***App E Op Code*** in set {E, M, U})

App E Calculated NOx Rate for Source = ***App E NOx MER***

else if (***App E Op Code*** == "W" AND ***Maximum App E Curve NOx Emission Rate*** is not null AND ***App E Reported Value*** >= 0 AND ***App E Reported Value*** is rounded to three decimal places)

If (***App E Reported Value*** >= ***Maximum App E Curve NOx Emission Rate*** * 1.25 (rounded to 3 decimal places))
App E Calculated NOx Rate for Source = ***App E Reported Value***

else
Count active MonitoringDefault record for location where
ParameterCode = "NORX" AND
DefaultPurpose = "MD" AND
FuelCode = ***App E Fuel Code***

if (*Count* <> 1)
return result A
else
NOx MER Default Record = matching record

If (*NOx MER Default Record*.DefaultValue > 0)
If (***App E Reported Value*** >= *NOx MER Default Record*.DefaultValue)
App E Calculated NOx Rate for Source = ***App E Reported Value***
else
if (***App E Reporting Method*** == "CONSTANT" or "APPORTIONED")
return result B
else
NOXR App E Accumulator = -1
return result C
else
if (***App E Reporting Method*** == "CONSTANT" or "APPORTIONED")
return result D
else
NOXR App E Accumulator = -1
return result D

if (***App E Calculated NOx Rate for Source*** is not null)

NOXR HPFF Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "NOXR" AND
UOM = "LBMMBTU"

if (***App E Reporting Method*** == "CONSTANT")

if (***App E Reported Value*** >= 0 AND ABS(***App E Calculated NOx Rate for Source - App E Reported Value***) > *NOXR HPFF Tolerance*)
return result E

if (***App E Reporting Method*** == "APPORTIONED")

Apportionment Calc NOXR Array at this Location = ***App E Calculated NOx Rate for Source***

if (***Rpt Period NOx Rate Calculated Accumulator Array*** for this location is not null)
if (***Rpt Period NOx Rate Calculated Accumulator Array*** for this location >= 0)
Rpt Period NOx Rate Calculated Accumulator Array for this location = ***Rpt Period NOx Rate Calculated Accumulator Array*** for this location + ***App E Calculated NOx Rate for Source***
else
Rpt Period NOx Rate Calculated Accumulator Array for this location = ***App E Calculated NOx Rate for Source***

Rpt Period NOx Rate Hours Accumulator Array for this location = ***Rpt Period NOx Rate Hours Accumulator Array*** for this location + 1
Set ***Current Measure Code*** to the ***Monitor Measure Code Array*** for "NOXR".

if (***App E Reported Value*** >= 0 AND ABS(***App E Calculated NOx Rate for Source - App E Reported Value***) > *NOXR HPFF Tolerance*)
return result E

else

If ***Current Fuel Flow Record***.FuelUsageTime > 0 AND ***Current Fuel Flow Record***.FuelUsageTime <= 1 AND ***NOXR App E Accumulator*** >= 0 AND ***App E Calc HI*** is not null)
NOXR App E Accumulator = ***NOXR App E Accumulator*** + (***App E Calculated NOx Rate for Source*** * ***Current Fuel Flow Record***.FuelUsageTime * ***App E Calc HI***)
else
NOXR App E Accumulator = -1

if (***App E Reported Value*** >= 0 AND ABS(***App E Calculated NOx Rate for Source*** - ***App E Reported Value***) > *NOXR HPFF Tolerance*)
return result F

else

if (***App E Reporting Method*** == "CONSTANT")
return result G

else if (***App E Reporting Method*** == "APPORTIONED")
Apportionment Calc NOXR Array at this Location = -1
Rpt Period NOX Rate Calculated Accumulator Array for this location = -1
return result G

else if (***App E Op Code*** is not null)
NOXR App E Accumulator = -1
return result H

Results:

Result	Response	Severity
A	The NOx emission rate could not be determined, because you did not report one and only one missing data default record for NORX for FuelCode [fuelcd] in your monitoring plan that was active during current hour.	Critical Error Level 1
B	You reported an OperatingConditionCode of W in the DHV record for NOXR for FuelCode [fuelcd], but the AdjustedHourlyValue is less than the minimum allowable substitute data value according to Appendix E sec. 2.5.2.1.	Critical Error Level 1
C	You reported an OperatingConditionCode of W in the HPFF record for NOXR for FuelCode [fuelcd], but the ParameterValueForFuel is less than the minimum allowable substitute data value according to Appendix E sec. 2.5.2.1.	Critical Error Level 1
D	The NOx emission rate could not be determined, because the DefaultValue in the NORX default record for FuelCode [fuelcd] is invalid.	Critical Error Level 1
E	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1
F	The ParameterValueForFuel reported in the HPFF record for NOXR for FuelCode [fuelcd] is inconsistent with the value recalculated from the Appendix E curve.	Critical Error Level 1
G	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated due to errors listed above.	Informational Message
H	The ParameterValueForFuel in the HPFF record for [parameter] for FuelCode [fuelcd] could not be recalculated due to errors listed above.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation
	Conditions:	App E Checks Needed Equals true
2	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification
	Conditions:	App E Constant Fuel Mix Equals true
3	Process/Category:	Emissions Data Evaluation Report --------- Hourly Fuel Flow
	Conditions:	App E Constant Fuel Mix Equals false

Check Code: HOURAE-13

Check Name: Check Reported NOx Emission Rate

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current Appe E NOXR Record*** is not null)

If (***Current App E NOXR Record***.ParamValFuel is null or is less than 0
return result A

else if (***Current App E NOXR Record***.ParamValFuel is not rounded to three decimal places)
return result B

Results:

Result	Response	Severity
A	The ParameterValueforFuel reported in the HPFF record for NOXR for FuelCode [fuelcd] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
B	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAE-14

Check Name: Check NOXR Units Of Measure

Related Former Checks:

Applicability: Appendix D Check

Description:

Specifications:

If (***Current App E NOXR Record*** is not null)

If (***Current App E NOXR Record***.ParameterUOMCode <> "LBMMBTU")
return result A

Results:

Result	Response	Severity
A	The ParameterUOMCode reported in the HPFF record for NOXR for FuelCode [fuelcd] is missing or invalid. The value should be "LBMMBTU".	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Code: HOURAE-15

Check Name: Determine Appendix E Measure Code

Related Former Checks:

Applicability: Appendix E Check

Description:

Specifications:

if (***App E Op Code*** is not null)
    If (***App E Op Code*** = "E" or ***Monitor Measure Code Array*** for "NOXR" =="OTHER")
        set ***Monitor Measure Code Array*** for "NOXR" to "OTHER"
    else if (***App E Op Code*** in set {M, U, N})
        if (***Monitor Measure Code Array*** for "NOXR" begins with "MEAS")
            set ***Monitor Measure Code Array*** for "NOXR" to "MEASSUB"
        else
            set ***Monitor Measure Code Array*** for "NOXR" to "SUB"
    else if (***App E Op Code*** in set {W, X, Y, Z})})
        if (***Monitor Measure Code Array*** for "NOXR" contains "SUB")
            set ***Monitor Measure Code Array*** for "NOXR" to "MEASSUB"
        else
            set ***Monitor Measure Code Array*** for "NOXR" to "MEASURE"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation
	Conditions:	App E Checks Needed Equals true
2	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification
	Conditions:	App E Constant Fuel Mix Equals true
3	Process/Category:	Emissions Data Evaluation Report --------- Hourly Fuel Flow
	Conditions:	App E Constant Fuel Mix Equals false

Check Code: HOURAE-16

Check Name: Update System Supplemental Data for Appendix E NOXE Hourly Param Fuel Flow System

Related Former Checks:

Applicability: Appendix E Check

Description:

Specifications:

If ***DerivedHourlyChecksNeeded*** is true AND ***CurrentOperatingTime*** is greater than 0,

If ***CurrentAppENoxrRecord***.MonitoringSystemID is not null,

Set *SupplementalDataDictionary* to the dictionary at ***CurrentMonitorPlanLocationPosition*** in ***SystemOperatingSuppDataDictionaryArray***.

If *SupplementalDataDictionary* contains key ***CurrentAppENoxrRecord***.MonitoringSystemID,

Set *SupplementalDataRecord* in *SupplementalDataDictionary* value where key is equal to ***CurrentAppENoxrRecord***.MonitoringSystemID.

Else

Create a new *SupplementalDataRecord* with MonitoringSystemID equal to ***CurrentAppENoxrRecord***.MonitoringSystemID, and OpDays, OpHours, OsDays and OsHours equal to 0..
Add *SupplementalDataRecord* to *SupplementalDataDictionary* with a key of ***CurrentAppENoxrRecord***.MonitoringSystemID.

Increament *SupplementalDataRecord*.QuarterlyOperatingCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.QuarterlyOperatingCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.MayAndJuneOperatingCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.

Increament *SupplementalDataRecord*.MayAndJuneOperatingCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.

Results:

Result	Response	Severity

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hourly Fuel Flow

Check Category:

Hourly Apportionment

Check Code: HOURAPP-1

Check Name: Determine Monitoring Plan Configuration

Related Former Checks: HOUROP-28

Applicability: CEM Check

Description: Determines whether the current MP is a Common Stack, Multistack, or simple Unit

Specifications:

MP Stack Config for Hourly Checks = null
MP Pipe Config for Hourly Checks = null
MP Load UOM = null
MP Unit Load = null
Stack OpTime Accumulator = 0
Stack LoadTimesOpTime Accumulator = 0
Stack HeatInputTimesOpTime Accumulator = 0
Pipe LoadTimesOpTime Accumulator = 0
Config HeatInputTimesOpTime Accumulator = 0
Config NOxRateTimesHeatInput Accumulator = 0
Config NOxRateTimesOpTime Accumulator = 0
Config OpTime Accumulator = 0
Config HeatInput Accumulator = 0
Max Stack OpTime = 0
Unit OpTime Accumulator = 0
Unit LoadTimesOpTime Accumulator = 0
Unit HeatInputTimesOpTime Accumulator = 0
Max Unit OpTime = 0
CP Fuel Count = 0
Current Month = month from ***Current Date***
App E Reporting Method = null
App E Op Code = null
App E Segment Number = null
App E Reported Value = null
App E Fuel Code = null
App E Calc HI = null
App E NOXE System ID = null
App E NOXE System Identifier = null
CurrentAppendixEStatus = null
EarliestLocationReportDate = null
Current Measure Code = null
MATS MS1 Hg DHV ID = null
MATS MS1 HCL DHV ID = null
MATS MS1 HF DHV ID = null
MATS MS1 SO2 DHV ID = null
MATS Parameter Plugin Hg = null
MATS Parameter Plugin HCL = null
MATS Parameter Plugin HF = null
MATS Parameter Plugin SO2 = null
MATS MS1 Hg Unadjusted Hourly Value = null
MATS MS1 HCL Unadjusted Hourly Value = null
MATS MS1 HF Unadjusted Hourly Value = null
MATS MS1 SO2 Unadjusted Hourly Value = null
Set ***Monitor Measure Code Array*** to null for each parameter.

For each array below, initialize each array with ***Current Location Count*** entries and the values as described

Apportionment OpTime Array - set each element in array to 0.0
Apportionment Load Array - set each element in array to 0

Apportionment Calc HI Array - set each element in array to 0.0
Apportionment Calc NOXR Array - set each element in array to 0.0
Apportionment HI Method Array - set each element in array to null
Apportionment NOX Method Array - set each element in array to null
Apportionment HI Measure Code Array - set each element in array to null
Apportionment NOXR Measure Code Array - set each element in array to null
Apportionment Stack Unit List - set each element in array to null
Apportionment NOXR Method Array - set each element in array to null
Apportionment Stack Flow Array - set each element in array to null
Apportionment MATS Load Array - set each element in array to null
Apportionment Hg Rate Array - set each element in array to null
Apportionment HCL Rate Array - set each element in array to null
Apportionment HF Rate Array - set each element in array to null
Apportionment SO2 Rate Array - set each element in array to null
MATS MS1 Hg MODC Code Array - set each element in array to null
MATS MS1 HCL MODC Code Array - set each element in array to null
MATS MS1 HF MODC Code Array - set each element in array to null
MATS MS1 SO2 MODC Code Array - set each element in array to null

If ***Current Location Count*** > 1

Find *List* of MonitorLocationIds in ***MonitorPlanLocation*** Table that match Current Monitoring Plan Id

For each MonitorLocationId in list, lookup record in ***MonitorLocation*** table
if StackPipeId is not null, add StackPipeId to *StackPipe list*
if UnitId is not null, add UnitId to *Unit list*

MS Count = 0
MP Count = 0
CS Count = 0
CP Count = 0
Unit Count = 0
CS Unit Count = 0
CP Unit Count = 0
Unit MS Count = 0

For each MonitorLocationId in list, lookup record in ***MonitorLocation*** table

if *MonitorLocation*.StackPipeID is not null,

set *Stack Unit Count* to 0

for each ***UnitStackConfiguration*** record where
BeginDate <= ***Current Date*** AND EndDate >= ***Current Date*** AND
StackPipeId = *MonitorLocation.StackPipeId*

add 1 to *Stack Unit Count*
append MonitorLocationID of the unit *to* ***Apportionment Stack Unit List*** for the stack location

if (*StackPipeUnit Count* > 0)
if (*MonitorLocation*.StackPipeName begins with "MS")
add 1 to *MS Count*

else if (*MonitorLocation*.StackPipeName begins with "MP")
add 1 to *MP Count*

else if (*MonitorLocation*.StackPipeName begins with "CS")
add 1 to *CS Count*

if (CS Count == 1)
CS Unit Count = Stack Unit Count

else if (*MonitorLocation*.StackPipeName begins with "CP")
add 1 to *CP Count*

if (*CP Count* == 1)
CP Unit Count = Stack Unit Count

else if UnitId is not null
add 1 to *Unit Count*

if (*Unit Count* == 1)
Unit MS Count = number of ***UnitStackConfiguration*** records where
UnitStackConfiguration.BeginDate <= ***Current Date*** AND
UnitStackConfiguration.EndDate >= ***Current Date*** AND
UnitStackConfiguration.UnitID = *MonitorLocation.UnitId*
UnitStackConfiguration.StackPipeName begins with "MS"

if (*MS Count* > 1 AND *CS Count* == 0 AND *Unit Count* == 1 AND *MS Count* == *Unit MS Count*)
MP Stack Config for Hourly Checks = "MS"
Multiple Stack Configuration = true
else if (*CS Count* == 1 AND *MS Count* == 0 AND *Unit Count* > 1 AND *Unit Count* == *CS Unit Count*)
MP Stack Config for Hourly Checks = "CS"
else if (*CS Count* == 1 AND *MS Count* > 0)
MP Stack Config for Hourly Checks = "CSMS"
else if (*CS Count* + *MS Count* > 0)
MP Stack Config for Hourly Checks = "COMPLEX"

If (*CP Count* ==1 AND *MP Count* == 0 AND *Unit Count* > 1 AND *Unit Count* == *CP Unit Count*)
MP Pipe Config for Hourly Checks = "CP"
else if (*CP Count* + *MP Count* > 0)
MP Pipe Config for Hourly Checks = "MULTIPLE"

Results:

Result	Response	Severity

Usage:

1 Process/Category: Emissions Data Evaluation Report --- Hourly Configuration Initialization

Check Code: HOURAPP-2

Check Name: Pre-Validate Heat Input Calculation

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Calculate Apportioned HI = false
Calculate NOXM From Apportioned HI = false

if ***Current Monitor Plan Location Record***.StackPipeID is not null
    If the StackPipeID of the monitoring location begins with "CS",
        set ***Current HI Entity Type*** = "CS"

    If the StackPipeID of the monitoring location begins with "CP",
        set ***Current HI Entity Type*** = "CP"

    If the StackPipeID of the monitoring location begins with "MS",
        set ***Current HI Entity Type*** = "MS"

    If the StackPipeID of the monitoring location begins with "MP",
        set ***Current HI Entity Type*** = "MP"

else if the UnitID of the monitoring location is not null
    set ***Current HI Entity Type*** = "Unit"

If (***Apportionment HI Method Array*** for the location contains "CALC") OR (***Apportionment HI Method Array*** for the location == "COMPLEX")

    If (***MP Pipe Config for Hourly Checks*** == "CP" AND ***CP Fuel Count*** > 1)
        Apportionment HI Method Array for the location == "NOCALC"

    If (***Apportionment OpTime Array*** for the location > 0 AND ***Apportionment OpTime Array*** for the location <= 1)

        // F-25
        If (***MP Stack Config for Hourly Checks*** == "CS" AND ***Current HI Entity Type*** = "CS")

            if (***Apportionment NOX Method Array*** == "NOXR")
                Calculate NOXM From Apportioned HI = true

            If ***Apportionment OpTime Array*** for the location < ***Max Unit OpTime***
                return result A
            else if ***Apportionment OpTime Array*** for the location > ***Unit OpTime Accumulator*** + (the number of units in the monitoring plan * .005)
                return result B
            else
                If (***Config HeatInputTimesOpTime Accumulator*** > 0)
                    Calculate Apportioned HI = true

                if (***MP Load UOM*** <> "INVALID" AND ***Stack LoadTimesOpTime Accumulator*** > 0 AND ***Unit LoadTimesOpTime Accumulator*** > 0 AND abs(***Stack LoadTimesOpTime Accumulator - Unit LoadTimesOpTime Accumulator***) >= number of items in the ***Apportionment OpTime Array***)
                    return result C

        // F-21A/B
        else if ((***MP Stack Config for Hourly Checks*** == "CS" OR ***MP Pipe Config for Hourly Checks*** == "CP") AND

Apportionment HI Method Array for the location not in set {NOCALC, COMPLEX})

if (***Apportionment NOX Method Array*** == "NOXR")
Calculate NOXM From Apportioned HI = true

If ***Max Stack OpTime*** < ***Max Unit OpTime*** AND ***MP Pipe Config for Hourly Checks*** is null
return result A
else if ***Max Stack OpTime*** > ***Unit OpTime Accumulator*** + (the number of units in the monitoring plan * .005) AND ***MP Pipe Config for Hourly Checks*** is null
return result B
else if ***MP Load UOM*** <> "INVALID"

if ((***MP Pipe Config for Hourly Checks*** <> "CP" AND ***MP Stack Config for Hourly Checks*** == "CS" AND ***Stack LoadTimesOpTime Accumulator*** > 0 AND ***Unit LoadTimesOpTime Accumulator*** > 0 AND (***Stack LoadTimesOpTime Accumulator*** - ***Unit LoadTimesOpTime Accumulator***) > number of items in the ***Apportionment OpTime Array***)
return result C

else if (***Config HeatInputTimesOpTime Accumulator*** >= 0 AND ***Apportionment Load Array*** for this Location >= 0 AND ***Unit LoadTimesOpTime Accumulator*** >= 0 AND ***Apportionment Calc HI Array*** for this Location >= 0)
Calculate Apportioned HI = true
If (***Unit LoadTimesOpTime Accumulator*** == 0)
Apportionment HI Method Array for the location == "NOCALC"

// Cannot apportionment but will validate total configuration
else if ((***MP Stack Config for Hourly Checks*** begins with "CS" OR ***MP Pipe Config for Hourly Checks*** == "CP" OR ***MP Pipe Config for Hourly Checks*** == "MULTIPLE") AND ***Apportionment HI Method Array*** for the location <> "COMPLEX")

if (***Apportionment NOX Method Array*** == "NOXR")
Calculate NOXM From Apportioned HI = true

If ***Max Stack OpTime*** > ***Unit OpTime Accumulator*** + (the number of units in the monitoring plan * .005) AND ***MP Pipe Config for Hourly Checks*** is null
return result B
else if (***Config HeatInputTimesOpTime Accumulator*** > 0 and ***Unit HeatInputTimesOpTime Accumulator*** == 0) OR (***Config HeatInputTimesOpTime Accumulator*** == 0 AND ***Unit HeatInputTimesOpTime Accumulator*** > 0)
return result G
else if (***Config HeatInputTimesOpTime Accumulator*** >= 0 AND ***Unit HeatInputTimesOpTime Accumulator*** >= 0)
Calculate Apportioned HI = true

// COMPLEX
else if (***MP Stack Config for Hourly Checks*** == "COMPLEX" OR ***Apportionment HI Method Array*** for the location == "COMPLEX")
if (***Apportionment NOX Method Array*** == "NOXR")
Calculate NOXM From Apportioned HI = true

if (***Config HeatInputTimesOpTime Accumulator*** > 0 and ***Unit HeatInputTimesOpTime Accumulator*** == 0) OR (***Config HeatInputTimesOpTime Accumulator*** == 0 AND ***Unit HeatInputTimesOpTime Accumulator*** > 0)
return result G

else if (***Config HeatInputTimesOpTime Accumulator*** >= 0 AND ***Unit HeatInputTimesOpTime Accumulator*** >= 0)
Calculate Apportioned HI = true

// F-21C
else if (***MP Stack Config for Hourly Checks*** == "MS")
    if (***Apportionment NOX Method Array*** == "NOXR")
        Calculate NOXM From Apportioned HI = true

    if (***Config HeatInputTimesOpTime Accumulator*** >= 0)
        Calculate Apportioned HI = true

    If ***Apportionment OpTime Array*** for the location < ***Max Stack OpTime***
        return result D
    else if ***Apportionment OpTime Array*** for the location > ***Stack OpTime Accumulator***
        return result E

else if (***Current Entity Type*** <>"Unit" AND ***Apportionment OpTime Array*** for the location > 0 AND the sum of ***Apportionment OpTime Array*** for all units in the ***Apportionment Stack Unit List*** for the location == 0)
    if (***Current Entity Type*** starts with "C")
        return result B
    else
        return result D

else if (***Current Entity Type*** == "MS" AND ***MP Load UOM*** <> "INVALID" and ***MP Unit Load*** > 0 AND ***Apportionment Load Array*** for the location > 0)
    if (***MP Unit Load*** <> ***Apportionment Load Array*** for the location)
        return result F

Results:

Result	Response	Severity
A	The OperatingTime reported at the common stack/pipe is less than the OperatingTime reported for one or more units linked to the stack/pipe.	Critical Error Level 1
B	The OperatingTime reported for one (or more) stacks/pipes is greater than the sum of the operating times reported for the units for the hour.	Critical Error Level 1
C	The HourLoad reported at the common stack/pipe is inconsistent with the load and operating time values reported at the units linked to the stack/pipe.	Critical Error Level 1
D	The OperatingTime reported for the unit is less than the OperatingTime reported for one or more multiple stacks linked to the unit.	Critical Error Level 1
E	The OperatingTime reported for the unit is greater than the sum of operating times at the multiple stacks linked to the unit.	Critical Error Level 1
F	The HourLoad in the Hourly Operating record for all multiple stacks linked to this unit are not equal.	Critical Error Level 1
G	The Heat Input Rate and Operating Time reported for the unit is inconsistent with the Heat Input Rates and Operating Times for the configuration.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation

Check Code: HOURAPP-3

Check Name: Calculate Apportioned or Summed Heat Input Rate

Related Former Checks: HOUROP-29

Applicability: CEM Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

Current HI Apportionment Record = null
HI Calculated Apportioned Value = null
App E Checks Needed = false

If (***Apportionment HI Method Array*** for the location contains "CALC" OR ***Apportionment HI Method Array*** for the location equals "COMPLEX")

If (***Calculate Apportioned HI*** = true)

Count active DerivedHourlyValueData records for location
WHERE ParameterCode = "HI"

If (*Count* == 1)
Current HI Apportionment Record = matching record

Heat Input Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "HI" AND
UOM = "MMBTUHR"
else
Calculate Apportioned HI = false

// F-25
If (***MP Stack Config for Hourly Checks*** == "CS" AND ***Current HI Entity Type*** = "CS")

If (***Calculate Apportioned HI*** == true)

HI Calculated Apportioned Value = ***Config HeatInputTimesOpTime Accumulator*** / ***Apportionment OpTime Array*** for this Location, rounded to one decimal place.

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
If (***Rpt Period HI Calculated Accumulator*** for this location >= 0)
Rpt Period HI Calculated Accumulator for this location = ***Rpt Period HI Calculated Accumulator*** for this location + (***HI Calculated Apportioned Value*** * ***Apportionment OpTime Array*** for this Location)

if (***Current Month*** is April)
April HI Calculated Accumulator for this location = ***April HI Calculated Accumulator*** for this location + (***HI Calculated Apportioned Value*** * ***Apportionment OpTime Array*** for this Location)

if (***Current HI Apportionment Record***.AdjustedHourlyValue >= 0 AND ABS(***Current HI Apportionment Record***.AdjustedHourlyValue - ***HI Calculated Apportioned Value***) > *Heat Input Tolerance*)
return result A

else if (***Apportionment OpTime Array*** for the location <> 0)
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period HI Calculated Accumulator for this location = -1

return result B

// other complex situations
else if (***Current HI Entity Type*** <> "Unit")

If (***Calculate Apportioned HI*** == true AND ***Current HI Apportionment Record***.AdjustedHourlyValue >= 0)

HI Calculated Apportioned Value = ***Current HI Apportionment Record***.AdjustedHourlyValue

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
If (***Rpt Period HI Calculated Accumulator*** for this location >= 0)
Rpt Period HI Calculated Accumulator for this location = ***Rpt Period HI Calculated Accumulator*** for this location + (***HI Calculated Apportioned Value * Apportionment OpTime Array*** for this Location)

if (***Current Month*** is April)
April HI Calculated Accumulator for this location = ***April HI Calculated Accumulator*** for this location + (***HI Calculated Apportioned Value * Apportionment OpTime Array*** for this Location)

else if (***Apportionment OpTime Array*** for the location <> 0)
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period HI Calculated Accumulator for this location = -1
return result B

// F-21A/B
else if ((***MP Stack Config for Hourly Checks*** == "CS" OR ***MP Pipe Config for Hourly Checks*** == "CP") AND ***(Apportionment HI Method Array*** for the location not in set {NOCALC, COMPLEX})

If (***Apportionment NOXR Method Array*** for the location = "AE")
App E Checks Needed = true

If (***Calculate Apportioned HI*** == true)

if (***Unit LoadTimesOpTime Accumulator*** > 0 OR ***Current HI Apportionment Record***.AdjustedHourlyValue >= 0)

if (***Unit LoadTimesOpTime Accumulator*** > 0)

HI Calculated Apportioned Value = (***Config HeatInputTimesOpTime Accumulator* Apportionment OpTime Array*** for this Location * ***Apportionment Load Array*** for this Location / ***Unit LoadTimesOpTime Accumulator***) / ***Apportionment OpTime Array*** for this Location), rounded to one decimal place.

else

HI Calculated Apportioned Value = ***Current HI Apportionment Record***.AdjustedHourlyValue

HI Calculated Apportioned Value = ***HI Calculated Apportioned Value*** + ***Apportionment Calc HI Array*** for this Location

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
If (***Rpt Period HI Calculated Accumulator*** for this location >= 0)
Rpt Period HI Calculated Accumulator for this location = ***Rpt Period HI Calculated Accumulator*** for this location + (***HI Calculated Apportioned Value * Apportionment OpTime Array*** for this Location)

if (***Current Month*** is April)

April HI Calculated Accumulator for this location = ***April HI Calculated Accumulator*** for this location + (***HI Calculated Apportioned Value * Apportionment OpTime Array*** for this Location)

if (***Current HI Apportionment Record***.AdjustedHourlyValue >= 0 AND ABS(***Current HI Apportionment Record***.AdjustedHourlyValue - ***HI Calculated Apportioned Value***) > *Heat Input Tolerance*)
return result A

else if (***Apportionment OpTime Array*** for the location <> 0)
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period HI Calculated Accumulator for this location = -1
return result B

else if (***Apportionment OpTime Array*** for the location <> 0)
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period HI Calculated Accumulator for this location = -1
return result B

// Cannot apportion or Complex configuration
else if (***MP Stack Config for Hourly Checks*** begins with "CS" OR ***MP Stack Config for Hourly Checks*** == "COMPLEX" OR ***MP Pipe Config for Hourly Checks*** in set {CP, MULTIPLE})

If (***Apportionment NOXR Method Array*** for the location = "AE")
App E Checks Needed = true

If (***Calculate Apportioned HI*** == true)

If (ABS(***Config HeatInputTimesOpTime Accumulator - Unit HeatInputTimesOpTime Accumulator***) <= *Heat Input Tolerance* OR ***Apportionment HI Method Array*** for the location == "COMPLEX" OR (***MP Stack Config for Hourly Checks***== "COMPLEX" and ***MP Pipe Config for Hourly Checks*** is null))

HI Calculated Apportioned Value = ***Current HI Apportionment Record***.AdjustedHourlyValue

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
If (***Rpt Period HI Calculated Accumulator*** for this location >= 0)
Rpt Period HI Calculated Accumulator for this location = ***Rpt Period HI Calculated Accumulator*** for this location + (***HI Calculated Apportioned Value * Apportionment OpTime Array*** for this Location)

if (***Current Month*** is April)
April HI Calculated Accumulator for this location = ***April HI Calculated Accumulator*** for this location + (***HI Calculated Apportioned Value * Apportionment OpTime Array*** for this Location)

else
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period HI Calculated Accumulator for this location = -1
return result C

else if (***Apportionment OpTime Array*** for the location <> 0)
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period HI Calculated Accumulator for this location = -1

Count active DerivedHourlyValueData records for location
WHERE ParameterCode = "HI"

If (*Count* == 1)
Current HI Apportionment Record = matching record
if (***Current HI Apportionment Record***.AdjustedHourlyValue > 0 AND ***Config HeatInputTimesOpTime Accumulator*** == 0)
return result D
else
return result B
else
return result B

// F-21C
else if (***MP Stack Config for Hourly Checks*** == "MS")

If (***Calculate Apportioned HI*** == true)

HI Calculated Apportioned Value = ***Config HeatInputTimesOpTime Accumulator*** / ***Unit OpTime Accumulator***, rounded to one decimal place.

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
If (***Rpt Period HI Calculated Accumulator*** for this location >= 0)
Rpt Period HI Calculated Accumulator for this location = ***Rpt Period HI Calculated Accumulator*** for this location + (***HI Calculated Apportioned Value * Unit OpTime Accumulator***)

if (***Current Month*** is April)
April HI Calculated Accumulator for this location = ***April HI Calculated Accumulator*** for this location + (***HI Calculated Apportioned Value * Apportionment OpTime Array*** for this Location)

if (***Current HI Apportionment Record***.AdjustedHourlyValue >= 0 AND ABS(***Current HI Apportionment Record***.AdjustedHourlyValue - ***HI Calculated Apportioned Value***) > *Heat Input Tolerance*)
return result A

else if (***Apportionment OpTime Array*** for the location <> 0)
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period HI Calculated Accumulator for this location = -1
return result B

Results:

Result	Response	Severity
A	The AdjustedHourlyValue reported in the DHV record for HI is inconsistent with the recalculated apportioned or summed value.	Critical Error Level 1
B	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated due to other errors listed in this report.	Informational Message
C	The heat input calculated for the configuration is inconsistent with the sum of the reported heat input at the units in this configuration.	Critical Error Level 1
D	You reported heat input at the unit, but there was no heat input at any of the locations where heat input was measured.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --- Hourly Configuration Evaluation

Check Code: HOURAPP-4

Check Name: Calculate NOx Mass Rate from Apportioned or Summed Heat Input Rate

Related Former Checks:

Applicability: CEM Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

Current NOX Apportionment Based Record = null
NOX Calculated Apportionment Based Value = null

If (***Calculate NOXM From Apportioned HI*** == true)

If (***HI Calculated Apportioned Value*** is not null AND ***Apportionment Calc NOXR Array*** for this location >= 0)

Count active DerivedHourlyValueData records for location and hour
WHERE ParameterCode = "NOX"

If (*Count* == 1)
Current NOX Apportionment Based Record = matching record

NOX Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "NOX" AND
UOM = "LBHR"

NOX Calculated Apportionment Based Value = HI Calculated Apportioned Value * Apportionment Calc NOXR Array, rounded to one decimal place.

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true

if (***Apportionment OpTime Array*** for this location is between 0 and 1 (inclusive))

If (***Rpt Period NOX Mass Calculated Accumulator*** for this location) >= 0)
Rpt Period NOX Mass Calculated Accumulator for this location = ***Rpt Period NOX Mass Calculated Accumulator*** for this location + (***NOX Calculated Apportionment Based Value * Apportionment OpTime Array*** for this location)

if (***Current Month*** is April)
April NOX Mass Calculated Accumulator for this location = ***April NOX Mass Calculated Accumulator*** for this location + (***NOX Calculated Apportionment Based Value * Apportionment OpTime Array*** for this location)

else
Rpt Period NOX Mass Calculated Accumulator for this location = -1

if (***Current NOX Apportionment Record***.AdjustedHourlyValue >= 0)
If (ABS(***Current HI Apportionment Record***.AdjustedHourlyValue - ***NOX Calculated Apportionment Based Value***) > *NOX Tolerance*)
If (***Legacy Data Evaluation*** == false)
return result A

else if (***Apportionment OpTime Array*** for this Location is greater than 0 and less than or equal to 1)

If (ABS(***Current HI Apportionment Record***.AdjustedHourlyValue - ***NOX Calculated Apportionment Based Value***) > *NOX Tolerance* / ***Apportionment OpTime Array*** for this Location)
return result A

else
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period NOX Mass Calculated Accumulator for this location = -1
return result B
else
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period NOX Mass Calculated Accumulator for this location = -1
return result B

Results:

Result	Response	Severity
A	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1
B	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated due to other errors listed in this report.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report --- Hourly Configuration Evaluation

Check Code: HOURAPP-5

Check Name: Sum Weighted NOx Emission Rate from Multiple Stacks

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

If (***MP Stack Config for Hourly Checks*** == "MS" AND ***Current HI Entity Type*** == "Unit")

If (***Config NOxRateTimesHeatInput Accumulator*** <> 0 OR ***Config NOxRateTimesOpTime Accumulator*** <> 0)

Expected Summary Value NOx Rate Array for this location = true

If (***Config NOxRateTimesHeatInput Accumulator*** > 0 AND ***Config HeatInput Accumulator*** > 0 AND ***Rpt Period NOX Rate Calculated Accumulator Array*** for this location >= 0)

Rpt Period NOX Rate Calculated Accumulator Array for this location = ***Rpt Period NOX Rate Calculated Accumulator*** for this location + (***Config NOxRateTimesHeatInput Accumulator*** / ***Config HeatInput Accumulator***, rounded to 3 decimal places.)

Rpt Period NOX Rate Hours Accumulator Array for this location = ***Rpt Period NOX Rate Hours Accumulator*** for this location + 1

else if (***Config NOxRateTimesOpTime Accumulator*** > 0 AND ***Config OpTime Accumulator*** > 0 AND ***Rpt Period NOX Rate Calculated Accumulator Array*** for this location >= 0)

Rpt Period NOX Rate Calculated Accumulator Array for this location = ***Rpt Period NOX Rate Calculated Accumulator*** for this location + (***Config NOxRateTimesOpTime Accumulator*** / ***Config OpTime Accumulator***, rounded to 3 decimal places.)

Rpt Period NOX Rate Hours Accumulator Array for this location = ***Rpt Period NOX Rate Hours Accumulator*** for this location + 1

else

Rpt Period NOX Rate Calculated Accumulator Array for this location = -1

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation

Check Code: HOURAPP-6

Check Name: Initialize Variable for Calculating Appendix E NOx Rate via Apportionment

Related Former Checks:

Applicability: Appendix E Check

Description:

Specifications:

If (***App E Checks Needed*** == true)
App E Op Code = null
App E Reporting Method = "APPORTIONED"

Count active DerivedHourlyValueData records for location and hour
WHERE ParameterCode = "NOX"

If (*Count* == 1)
Current NOXR Apportionment Based Record = matching record

if (***Current NOXR Apportionment Based Record***.MonitoringSystemID is not null)

Mon Sys Record = find active MonitoringSystemData record for location where MonitoringSystemData.MonitoringSystemID = ***Current NOXR Apportionment Based Record***.MonitoringSystemID

if (found AND *Mon Sys Record*.SystemTypeCode == "NOXE" AND *Mon Sys Record*.FuelTypeCode is not null)

if (***Current NOXR Apportionment Based Record***.OperatingConditionCode in set {X, Y, Z, U, W, N, M})

App E Op Code = ***Current NOXR Apportionment Based Record.***OperatingConditionCode
App E Calc HI = ***HI Calculated Apportioned Value***
App E Reported Value = ***Current NOXR Apportionment Based Record***.AdjustedHourlyValue
App E Segment Number = ***Current NOXR Apportionment Based Record***.SegmentNumber
App E NOXE System ID = ***Current NOXR Apportionment Based Record***.MonitoringSystemID
App E NOXE System Identifier = ***Current NOXR Apportionment Based Record***.SystemIdentifier
App E Fuel Code = *Mon Sys Record*.Fuel Type Code
EarliestLocationReportDate = ***CurrentMonitorPlanLocationRecord***.EarliestReportDate

else if (***Current NOXR Apportionment Based Record***.OperatingConditionCode == "E")
return result A

else
return result B
else
return result C

Results:

Result	Response	Severity
A	You reported an OperatingConditionCode of E in the DHV record for NOXR. You should report the NOx emission rate for emergency fuels in an HPFF record, not a DHV record.	Critical Error Level 1
B	The OperatingConditionCode reported in the DHV record for NOXR is missing or invalid.	Critical Error Level 1
C	According to your monitoring plan, your reported that you are determining NOx emission rate using the Appendix E methodology, but you did not report a MonitoringSystemID in this record. You should report the MonitoringSystemID of the NOXE system associated with the Appendix E fuel curve.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation

Check Code: HOURAPP-7

Check Name: Handle NOx Rate Summary Expected for ARP

Related Former Checks:

Applicability:

Description: Sets Expected Summary Value NOx Rate Array value for a location to true when the location is a unit and the unit is affected by ARP.

Specifications:

If ***CurrentHIEntityType*** is equal to "Unit", ***MPStackConfigForHourlyChecks*** is equal to "MS", and the ***ExpectedSummaryValueNOxRateArray*** value for this location is equal to false,

Locate a record in ***LocationProgramRecordsByHourAndLocation*** where:

1) ProgramCode is equal to 'ARP', and
2) Class is equal to 'P1' or 'P2'.

if found,

Set ***ExpectedSummaryValueNOxRateArray*** for this location to true

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation

Check Code: HOURAPP-9

Check Name: Check MATS Load Value

Related Former Checks:

Applicability: General Check

Description: This checks the apportionment of load value to MS in MS configurations. The load is apportioned based on stack flow. The check calculates the apportioned value and compares it to the reported value with a tolerance determine by HourlyEmissionsTolerances's LOAD-MW entry.

Specifications:

Set ***CalculatedMatsMsLoad*** to null.

If ***MpStackConfigForHourlyChecks*** is equal to "MS", AND ***CurrentMonitorPlanLocationRecord***.StackPipeID is not null,

Set *CurrentMsLoad to* ***ApportionmentMatsLoadArray*** value for the current location.

If *CurrentMsLoad* is NOT null,

If the ***ApportionmentStackFlowArray*** value for every MS location is NOT null,

Set *MsStackFlowSum* to the sum of ***ApportionmentStackFlowArray*** * ***ApportionmentOpTimeArray*** for MS locations .
Set *CurrentMsFlow to* ***ApportionmentStackFlowArray*** * ***ApportionmentOpTimeArray*** for the current location.
Set *UnitLoad* to ***ApportionmentMatsLoadArray*** value for the unit.

If *MsStackFlowSum* is greater than 0, AND *UnitLoad* is NOT null and is greater than 0,

Set ***CalculatedMatsMsLoad*** to *UnitLoad* * *CurrentMsFlow* / *MsStackFlowSum*, rounded to an integer.

If ABS(*CurrentMsLoad* - ***CalculatedMatsMsLoad***) is greater than ***MwLoadHourlyTolerance***,

Return result A.

Results:

Result	Response	Severity
A	The reported MATS Load value does not match the value of [CALCVALUE] calculated using stack flow apportionment.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report --- Hourly Configuration Evaluation

Check Code: HOURAPP-10

Check Name: MATS Hg: Calculate and check MATS MS1 Flow Weighted value

Related Former Checks:

Applicability: General Check

Description: Calculates and checks the MATS rate when the source is using the MS-1 formula.

Specifications:

Set ***CalculatedFlowWeightedHg*** to null.
Set ***MATSReportedPluginHg*** to null.

If ***MpStackConfigForHourlyChecks*** is equal to "MS", AND ***CurrentMonitorPlanLocationRecord***.UnitID is not null, AND ***MatsHgDhvId*** is not null,
Set *Modc38Used* to false.
Set *StackOperated* to false.
Set *StackMissingData* to false.
Set *NumOperatingStacks* to 0.
Set *SingleStackHgRate* to 0.

For each MS Location
If ***MatsMS1HgModcCodeArray*** is 38
Set *Modc38Used* to true

If ***ApportionmentOpTimeArray*** is > 0
Set *StackOperated* to true
Increment *NumOperatingStacks*
Set *SingleStackHgRate* to ***ApportionmentHgRateArray***

If ***ApportionmentOpTimeArray*** is > 0 and (***ApportionmentStackFlowArray*** is null or ***ApportionmentHgRateArray*** is null),
Set *StackMissingData* to true

If the ***ApportionmentStackFlowArray*** value for MS location is NOT null,
Set *MsStackFlowSum* to the sum of ***ApportionmentStackFlowArray*** for MS location.
Set *MsStackEmissionRateFlow* to the sum of ***ApportionmentHgRateArray * ApportionmentStackFlowArray*** for MS location.

If *Modc38Used* is false and *StackOperated* is true and ***MatsHgUnadjustedHourlyValue*** is null
return result B
Else If *NumOperatingStacks* is 1
Set ***CalculatedFlowWeightedHg*** to *SingleStackHgRate*

If *SingleStackHgRate* not equal ***MatsHgUnadjustedHourlyValue***
return result A
Else If *StackMissingData* is true and ***MatsHgUnadjustedHourlyValue*** is not null
return result C
Else If *MsStackFlowSum* is greater than 0 and ***MatsHgUnadjustedHourlyValue*** is not null
Set ***CalculatedFlowWeightedHg*** to *MsStackEmissionRateFlow* / *MsStackFlowSum*, rounded to an integer.
Set ***MATSReportedPluginHg*** to ***MatsHgUnadjustedHourlyValue***

If (***MatsHgUnadjustedHourlyValue*** + ***CalculatedFlowWeightedHg***) is NOT equal to 0
Set *PercentDifference* = 100 * ABS(***MatsMs1HgUnadjustedHourlyValue - CalculatedFlowWeightedHg***)/ ((***MatsMs1HgUnadjustedHourlyValue*** + ***CalculatedFlowWeightedHg***)/2), **rounded to 1 decimal place.**

If (*PercentDifference* > 5)
return result A

Results:

Result	Response	Severity
A	The reported [MATS Parameter] Calculated Flow Weighted value of [Reported] does not match the recalculated value of [Calculated] calculated using stack flow apportionment.	Informational Message
B	You did not report a [MATS Parameter] unit-level emission rate, but did report a [MATS Parameter] emission rate at each operating stack.	Informational Message
C	You reported a [MATS Parameter] unit-level emission rate, but did not report a [MATS Parameter] emission rate or unadjusted flow rate from one or both stacks.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation

Check Code: HOURAPP-11

Check Name: MATS HCL: Calculate and check MATS MS1 Flow Weighted value

Related Former Checks:

Applicability: General Check

Description: Calculates and checks the MATS rate when the source is using the MS-1 formula.

Specifications:

Set ***CalculatedFlowWeightedHcl*** to null.
Set ***MATSReportedPluginHcl*** to null.

If ***MpStackConfigForHourlyChecks*** is equal to "MS", AND ***CurrentMonitorPlanLocationRecord***.UnitID is not null, AND ***MatsHclDhvId*** is not null,
Set *Modc38Used* to false.
Set *StackOperated* to false.
Set *StackMissingData* to false.
Set *NumOperatingStacks* to 0.
Set *SingleStackHclRate* to 0.

For each MS Location
If ***MatsMS1HgModcCodeArray*** is 38
Set *Modc38Used* to true

If ***ApportionmentOpTimeArray*** is > 0
Set *StackOperated* to true
Increment *NumOperatingStacks*
Set *SingleStackHclRate* to ***ApportionmentHclRateArray***

If ***ApportionmentOpTimeArray*** is > 0 and (***ApportionmentStackFlowArray*** is null or ***ApportionmentHclRateArray*** is null),
Set *StackMissingData* to true

If the ***ApportionmentStackFlowArray*** value for MS location is NOT null,
Set *MsStackFlowSum* to the sum of ***ApportionmentStackFlowArray*** for MS location.
Set *MsStackEmissionRateFlow* to the sum of ***ApportionmentHclRateArray*** * ***ApportionmentStackFlowArray*** for MS location.

If *Modc38Used* is false and *StackOperated* is true and ***MatsHclUnadjustedHourlyValue*** is null
return result B
Else If *NumOperatingStacks* is 1
Set ***CalculatedFlowWeightedHcl*** to *SingleStackHclRate*

If *SingleStackHclRate* not equal ***MatsHclUnadjustedHourlyValue***
return result A
Else If *StackMissingData* is true and ***MatsHclUnadjustedHourlyValue*** is not null
return result C
If *MsStackFlowSum* is greater than 0 and ***MatsHclUnadjustedHourlyValue*** is not null
Set ***CalculatedFlowWeightedHcl*** to *MsStackEmissionRateFlow* / *MsStackFlowSum*, rounded to an integer.
Set ***MATSReportedPluginHcl*** to ***MatsHclUnadjustedHourlyValue***

If (***MatsHclUnadjustedHourlyValue*** + ***CalculatedFlowWeightedHcl***) is NOT equal to 0
Set *PercentDifference* = 100 * ABS(***MatsMs1HclUnadjustedHourlyValue*** - ***CalculatedFlowWeightedHcl***)/ ((***MatsMs1HclUnadjustedHourlyValue*** + ***CalculatedFlowWeightedHcl***)/2), **rounded to 1 decimal place.**

If (*PercentDifference* > 5)
return result A

Results:

Result	Response	Severity
A	The reported [MATS Parameter] Calculated Flow Weighted value of [Reported] does not match the recalculated value of [Calculated] calculated using stack flow apportionment.	Informational Message
B	You did not report a [MATS Parameter] unit-level emission rate, but did report a [MATS Parameter] emission rate at each operating stack.	Informational Message
C	You reported a [MATS Parameter] unit-level emission rate, but did not report a [MATS Parameter] emission rate or unadjusted flow rate from one or both stacks.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation

Check Code: HOURAPP-12

Check Name: MATS HF: Calculate and check MATS MS1 Flow Weighted value

Related Former Checks:

Applicability: General Check

Description: Calculates and checks the MATS rate when the source is using the MS-1 formula.

Specifications:

Set ***CalculatedFlowWeightedHf*** to null.
Set ***MATSReportedPluginHf*** to null.

If ***MpStackConfigForHourlyChecks*** is equal to "MS", AND ***CurrentMonitorPlanLocationRecord***.UnitID is not null, AND ***MatsHfDhvId*** is not null,
Set *Modc38Used* to false.
Set *StackOperated* to false.
Set *StackMissingData* to false.
Set *NumOperatingStacks* to 0.
Set *SingleStackHfRate* to 0.

For each MS Location
If ***MatsMS1HgModcCodeArray*** is 38
Set *Modc38Used* to true

If ***ApportionmentOpTimeArray*** is > 0
Set *StackOperated* to true
Increment *NumOperatingStacks*
Set *SingleStackHfRate* to ***ApportionmentHfRateArray***

If ***ApportionmentOpTimeArray*** is > 0 and (***ApportionmentStackFlowArray*** is null or ***ApportionmentHfRateArray*** is null),
Set *StackMissingData* to true

If the ***ApportionmentStackFlowArray*** value for MS location is NOT null,
Set *MsStackFlowSum* to the sum of ***ApportionmentStackFlowArray*** for MS location.
Set *MsStackEmissionRateFlow* to the sum of ***ApportionmentHfRateArray*** * ***ApportionmentStackFlowArray*** for MS location

If *Modc38Used* is false and *StackOperated* is true and ***MatsHfUnadjustedHourlyValue*** is null
return result B
Else If *NumOperatingStacks* is 1
Set ***CalculatedFlowWeightedHf*** to *SingleStackHfRate*

If *SingleStackHfRate* not equal ***MatsHfUnadjustedHourlyValue***
return result A
Else If *StackMissingData* is true and ***MatsHfUnadjustedHourlyValue*** is not null
return result C
Else If *MsStackFlowSum* is greater than 0 and ***MatsHfUnadjustedHourlyValue*** is not null
Set ***CalculatedFlowWeightedHf*** to *MsStackEmissionRateFlow* / *MsStackFlowSum*, rounded to an integer.
Set ***MATSReportedPluginHf*** to ***MatsHfUnadjustedHourlyValue***

If (***MatsHfUnadjustedHourlyValue*** + ***CalculatedFlowWeightedHf***) is NOT equal to 0
Set *PercentDifference* = 100 * ABS(***MatsMs1HfUnadjustedHourlyValue*** - ***CalculatedFlowWeightedHf***)/ ((***MatsMs1HfUnadjustedHourlyValue*** + ***CalculatedFlowWeightedHf***)/2), **rounded to 1 decimal place.**

If (*PercentDifference* > 5)
return result A

Results:

Result	Response	Severity
A	The reported [MATS Parameter] Calculated Flow Weighted value of [Reported] does not match the recalculated value of [Calculated] calculated using stack flow apportionment.	Informational Message
B	You did not report a [MATS Parameter] unit-level emission rate, but did report a [MATS Parameter] emission rate at each operating stack.	Informational Message
C	You reported a [MATS Parameter] unit-level emission rate, but did not report a [MATS Parameter] emission rate or unadjusted flow rate from one or both stacks.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation

Check Code: HOURAPP-13

Check Name: MATS SO2: Calculate and check MATS MS1 Flow Weighted value

Related Former Checks:

Applicability: General Check

Description: Calculates and checks the MATS rate when the source is using the MS-1 formula.

Specifications:

Set ***CalculatedFlowWeightedSo2*** to null.
Set ***MATSReportedPluginSo2*** to null.

If ***MpStackConfigForHourlyChecks*** is equal to "MS", AND ***CurrentMonitorPlanLocationRecord***.UnitID is not null, AND ***MatsSo2DhvId*** is not null,
Set *Modc38Used* to false.
Set *StackOperated* to false.
Set *StackMissingData* to false.
Set *NumOperatingStacks* to 0.
Set *SingleStackSo2Rate* to 0.

For each MS Location
If ***MatsMS1HgModcCodeArray*** is 38
Set *Modc38Used* to true

If ***ApportionmentOpTimeArray*** is > 0
Set *StackOperated* to true
Increment *NumOperatingStacks*
Set *SingleStackSo2Rate* to ***ApportionmentSo2RateArray***

If ***ApportionmentOpTimeArray*** is > 0 and (***ApportionmentStackFlowArray*** is null or ***ApportionmentSo2RateArray*** is null),
Set *StackMissingData* to true

If the ***ApportionmentStackFlowArray*** value for MS location is NOT null,
Set *MsStackFlowSum* to the sum of ***ApportionmentStackFlowArray*** for MS location.
Set *MsStackEmissionRateFlow* to the sum of ***ApportionmentSo2RateArray*** * ***ApportionmentStackFlowArray*** for MS location.

If *Modc38Used* is false and *StackOperated* is true and ***MatsSo2UnadjustedHourlyValue*** is null
return result B
Else If *NumOperatingStacks* is 1
Set ***CalculatedFlowWeightedSo2*** to *SingleStackSo2Rate*

If *SingleStackSo2Rate* not equal ***MatsSo2UnadjustedHourlyValue***
return result A
Else If *StackMissingData* is true and ***MatsSo2UnadjustedHourlyValue*** is not null
return result C
Else If *MsStackFlowSum* is greater than 0 and ***MatsSo2UnadjustedHourlyValue*** is not null
Set ***CalculatedFlowWeightedSo2*** to *MsStackEmissionRateFlow* / *MsStackFlowSum*, rounded to an integer.
Set ***MATSReportedPluginSo2*** to ***MatsSo2UnadjustedHourlyValue***

If (***MatsSo2UnadjustedHourlyValue*** + ***CalculatedFlowWeightedSo2***) is NOT equal to 0
Set *PercentDifference* = 100 * ABS(***MatsMs1So2UnadjustedHourlyValue*** - ***CalculatedFlowWeightedSo2***)/ ((***MatsMs1So2UnadjustedHourlyValue*** + ***CalculatedFlowWeightedSo2***)/2), **rounded to 1 decimal place.**

If (*PercentDifference* > 5)
return result A

Results:

Result	Response	Severity
A	The reported [MATS Parameter] Calculated Flow Weighted value of [Reported] does not match the recalculated value of [Calculated] calculated using stack flow apportionment.	Informational Message
B	You did not report a [MATS Parameter] unit-level emission rate, but did report a [MATS Parameter] emission rate at each operating stack.	Informational Message
C	You reported a [MATS Parameter] unit-level emission rate, but did not report a [MATS Parameter] emission rate or unadjusted flow rate from one or both stacks.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Hourly Configuration Evaluation

Check Code: HOURAPP-14

Check Name: Validate Complex Configuration HI

Related Former Checks:

Applicability: General Check

Description: Ensures that the heat input reported at the stacks for a complex configuration with stacks are consistent by ensuring that the total heat input reported at the units equals the toral reported at the stacks for each hour.

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

If (***MpStackConfigForHourlyChecks*** == "COMPLEX" or "CSMS")

If (***StackHeatInputTimesOpTimeAccumulator*** >= 0) AND (***UnitHeatInputTimesOpTimeAccumulator*** >= 0)

HeatInputTolerance = Tolerance from ***HourlyEmissionsToleranceCrossCheckTable*** where Parameter == "HI" and Uom == "MMBTUHR".

If (ABS(***StackHeatInputTimesOpTimeAccumulator*** - ***UnitHeatInputTimesOpTimeAccumulator***) > (*HeatInputTolerance* * ***CurrentLocationCount***))

return result A.

Results:

Result	Response	Severity
A	The heat input reported at the units is not consistent with the heat input reported at the stacks in this configuration.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --- Hourly Apportionment Verification

Check Category:

Hourly Calculated Data

Check Code: HOURCV-1

Check Name: Calculate Percent H2O

Related Former Checks:

Applicability: CEM Check

Description: If there was a valid equation code reported for a Derived H2O record and O2 monitored values were available, this check reproduces the formula for calculating H2O and compares it to the reported H2O Percentage

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

if (***H2O Method Code*** = "MWD" AND ***Current DHV Record***.ModcCode in set {01, 02, 03, 04, 53, 54})

if (***H2O CEM Equation Code*** == "F-31")

if (***Current DHV Record Valid*** == true AND ***O2 Wet Calculated Adjusted Value*** is not null AND ***O2 Dry Calculated Adjusted Value*** is not null)

H2O DHV Calculated Adjusted Value = ((***O2 Dry Calculated Adjusted Value - O2 Wet Calculated Adjusted Value***) * 100.0) / ***O2 Dry Calculated Adjusted Value,*** ROUNDED to one decimal place.

H2O Conc Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "H2O" AND
UOM = "PCT"

If (***Derived Hourly Adjusted Value Status*** == true AND ABS(***H2O DHV Calculated Adjusted Value - Current DHV Record***.AdjustedHourlyValue) > H2O Conc Tolerance)
return result A

else
return result B

else if (***H2O CEM Equation Code*** == "M-1K")

if (***Derived Hourly Adjusted Value Status*** == true)
H2O DHV Calculated Adjusted Value = ***Current DHV Record***.AdjustedHourlyValue

else
return result B

else if (***H2O Method Code*** = "MDF" AND ***Current DHV Record***.ModcCode == "40")
H2O DHV Calculated Adjusted Value = ***H2O Default Value***

else
H2O DHV Calculated Adjusted Value = ***Current DHV Calculated Adjusted Value***

Results:

Result	Response	Severity
A	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1
B	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated due to errors listed above.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ H2O Calculation Verification

Check Code: HOURCV-3

Check Name: Determine Diluent Cap and Moisture for CO2 Concentration Calculation Verification

Related Former Checks:

Applicability: CEM Check

Description: Determines the moisture and diluent values used in the CO2C calculation.

Specifications:

if (***CO2 Conc CEM Equation Code*** == "F-14B")
    If (***H2O Method Code*** == "MWD" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
        Calculated Moisture for CO2C = H2O DHV Calculated Adjusted Value
    else if (***H2O Method Code*** in set {MMS, MTB} AND ***H2O Monitor Hourly Checks Needed*** == true AND ***H2O MHV Calculated Adjusted Value*** is not null)
        Calculated Moisture for CO2C = H2O MHV Calculated Adjusted Value
    else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
        Calculated Moisture for CO2C = H2O DHV Calculated Adjusted Value
    else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == false AND ***H2O Default Value*** is not null)
        Calculated Moisture for CO2C = H2O Default Value

if (***Use O2 Diluent Cap for Co2 Conc Calc*** == true)
    O2X Count = # of active MonitoringDefault records for location where
        ParameterCode = 'O2X' AND DefaultPurposeCode = 'DC' AND
        FuelCode = 'NFS'
    if (*O2X Count* > 1)
        return result A
    else if (*O2X Count* == 0)
        return result B
    else if MonitoringDefault.DefaultValue <= 0
        return result C
    else
        Calculated Diluent for CO2C = MonitoringDefault.DefaultValue

else

    case (***CO2 Conc CEM Equation Code***)
        "F-14A":
            Calculated Diluent for CO2C = O2 Dry Calculated Adjusted Value
        "F-14B":
            Calculated Diluent for CO2C = O2 Wet Calculated Adjusted Value

Results:

Result	Response	Severity
A	You reported more than one diluent cap default record for O2X in your monitoring plan that was active during current hour.	Critical Error Level 1
B	You did not report a default record for O2X in your monitoring plan that was active during the current hour. Please note that the use of a diluent cap to calculate CO2 concentration is only applicable to legacy data.	Critical Error Level 1
C	The DefaultValue reported in the active Default record for O2X in your monitoring plan is invalid. The value must be greater than 0.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Calculation Verification

Check Code: HOURCV-4

Check Name: Calculate CO2 Concentration

Related Former Checks:

Applicability: CEM Check

Description: Based on equation code in CO2 Concentration record and reported values, calculate the CO2 Concentration

Specifications:

If (***Current DHV Record***.ModcCode in set {01, 02, 03, 04, 53, 54})

CO2 Conc Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerance" where
Parameter = "CO2C" AND
UOM = "PCT"

case (***CO2 Conc CEM Equation Code***)
"F-14A":

If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for CO2C*** is not null AND ***Valid FC Factor Exists*** == true AND ***Valid FD Factor*** == true)

CO2C DHV Calculated Adjusted Value = 100 * (***Current Hourly Op Record***.FcFactor /***Current Hourly Op Record***.FdFactor) * [(20.9 - ***Calculated Diluent for CO2C***) / 20.9], and round the result to 1 decimal place.

If (***CO2C DHV Calculated Adjusted Value*** < 0)
CO2C DHV Calculated Adjusted Value = 0

If (***Derived Hourly Adjusted Value Status*** == true AND ABS(***CO2C DHV Calculated Adjusted Value - Current DHV Record***.AdjustedHourlyValue) > CO2 Conc Tolerance)
return result A

else
return result B

"F-14B":
If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for CO2C*** is not null AND ***Valid FC Factor Exists*** == true AND ***Valid FD Factor Exists*** == true AND ***Calculated Moisture for CO2C*** is not null)

CO2C DHV Calculated Adjusted Value = [100/20.9] * (***Current Hourly Op Record***.FcFactor /***Current Hourly Op Record***.FdFactor) * [20.9 * ((100 - ***Calculated Moisture for CO2C***)/100) - ***Calculated Diluent for CO2C***], and round the result to 1 decimal place.

If (***CO2C DHV Calculated Adjusted Value*** < 0)
CO2C DHV Calculated Adjusted Value = 0

If (***Derived Hourly Adjusted Value Status*** == true AND ABS(***CO2C DHV Calculated Adjusted Value - Current DHV Record***.AdjustedHourlyValue) > CO2 Conc Tolerance)
return result A

else
return result B

Otherwise
return result B

else
CO2C DHV Calculated Adjusted Value = ***Current DHV Calculated Adjusted Value***

Results:

Result	Response	Severity
A	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1
B	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated due to errors listed above.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Calculation Verification

Check Code: HOURCV-6

Check Name: Determine Diluent Cap and Moisture for Heat Input Calculation Verification

Related Former Checks:

Applicability: General Check

Description: Ensures that all inputs are available for each equation type prior to performing the actual calculations

Specifications:

If (***Heat Input Method Code*** == "CEM")
    if (***Heat Input Equation Code*** in set {F-16, F-17, F-18}
        If (***H2O Method Code*** == "MWD" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
            Calculated Moisture for HI = H2O DHV Calculated Adjusted Value
        else if (***H2O Method Code*** in set {MMS, MTB} AND ***H2O Monitor Hourly Checks Needed*** == true AND ***H2O MHV Calculated Adjusted Value*** is not null)
            Calculated Moisture for HI = H2O MHV Calculated Adjusted Value
        else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
            Calculated Moisture for HI = H2O DHV Calculated Adjusted Value
        else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == false AND ***H2O Default Value*** is not null)
            Calculated Moisture for HI = H2O Default Value

    if (***Heat Input Equation Code*** = "F-15" OR ***Heat Input Equation Code*** = "F-16"
        If (***Current DHV Record***.DiluentCapIndicator == 1)
            CO2N Count = # of active MonitoringDefault records for the location where
                ParameterCode = 'CO2N' AND DefaultPurposeCode = 'DC' AND
                FuelCode = 'NFS'
            if (*CO2N Count* > 1)
                return result A
            else if (*CO2N Count* == 0)
                return result B
            else if MonitoringDefault.DefaultValue <= 0
                return result C
            else
                Calculated Diluent For HI = MonitoringDefault.DefaultValue

        else if (***CO2 Conc Checks Needed for Heat Input*** == true)

            if (***Current CO2 Conc Missing Data Monitor Hourly Record*** is not null)
                Calculated Diluent for HI = ***CO2C SD Calculated Adjusted Value***
            else
                Calculated Diluent for HI = ***CO2C MHV Calculated Adjusted Value***

    else if (***Heat Input Equation Code*** == "F-17" OR ***Heat Input Equation Code*** == "F-18")
        if (***Current DHV Record***.DiluentCapIndicator == 1)
            O2X Count = # of active MonitoringDefault records for the location where
                ParameterCode = 'O2X' AND DefaultPurposeCode = 'DC' AND
                FuelCode = 'NFS'
            if (*O2X Count* > 1)
                return result D
            else if (*O2X Count* == 0)
                return result E
            else if MonitoringDefault.DefaultValue <= 0
                return result F
            else
                Calculated Diluent For HI = MonitoringDefault.DefaultValue

else if (***Heat Input Equation Code*** == "F-17" AND ***O2 Wet Checks Needed for Heat Input*** == true)

if (***Current O2 Wet Missing Data Monitor Hourly Record*** is not null)
Calculated Diluent for HI = ***O2C SD Calculated Adjusted Value***
else
Calculated Diluent for HI = ***O2 Wet Calculated Adjusted Value***

else if (***Heat Input Equation Code*** == "F-18" AND ***O2 Dry Checks Needed for Heat Input*** == true)

if (***Current O2 Dry Missing Data Monitor Hourly Record*** is not null)
Calculated Diluent for HI = ***O2C SD Calculated Adjusted Value***
else
Calculated Diluent for HI = ***O2 Dry Calculated Adjusted Value***

Results:

Result	Response	Severity
A	You reported more than one diluent cap default record for CO2N in your monitoring plan that was active during the current hour.	Critical Error Level 1
B	You did not report an active CO2N diluent cap default record in your monitoring plan for the hour. The use of the diluent cap to calculate HI is only applicable for legacy data.	Critical Error Level 1
C	The DefaultValue reported in the active Default record for CO2N in your monitoring plan is invalid. The value must be greater than 0.	Critical Error Level 1
D	You reported more than one diluent cap default record for O2X in your monitoring plan that was active during current hour.	Critical Error Level 1
E	You did not report a default record for O2X in your monitoring plan that was active during the current hour. Please note that the use of a diluent cap to calculate HI is only applicable to legacy data.	Critical Error Level 1
F	The DefaultValue reported in the active Default record for O2X in your monitoring plan is invalid. The value must be greater than 0.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Heat Input Calculation Verification

Check Code: HOURCV-7

Check Name: Calculate Heat Input

Related Former Checks:

Applicability: General Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

if (***Derived Hourly Adjusted Value Status*** == true AND ***Current Hourly Op Record*****.**OperatingTime is between 0 and 1 (inclusive))

Heat Input Total Reported Value = ***Current DHV Record***.AdjustedHourlyValue * ***Current Hourly Op Record*****.**OperatingTime.
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
if (***Rpt Period HI Reported Accumulator Array*** for this location is not null)
if (***Rpt Period HI Reported Accumulator Array*** >= 0)
Rpt Period HI Reported Accumulator Array for this location = ***Rpt Period HI Reported Accumulator Array*** for this location + *Heat Input Total Reported Value*
else
Rpt Period HI Reported Accumulator Array for this location = *Heat Input Total Reported Value*

if (***Unit HeatInputTimesOpTime Accumulator*** >= 0)
if (***Current Entity Type*** == "Unit")
Unit HeatInputTimesOpTime Accumulator = ***Unit HeatInputTimesOpTime Accumulator*** + *Heat Input Total Reported Value*
elsi if (***Current Entity Type*** == "CS" or "MS")
Stack HeatInputTimesOpTime Accumulator = ***Stack HeatInputTimesOpTime Accumulator*** + *Heat Input Total Reported Value*

else
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period HI Reported Accumulator Array for this location = -1

if (***Current Entity Type*** == "Unit")
Unit HeatInputTimesOpTime Accumulator = -1
elsi if (***Current Entity Type*** == "CS" or "MS")
Stack HeatInputTimesOpTime Accumulator = -1

Total Heat Input from Fuel Flow = null

If (***Heat Input Method Code*** == "CEM")

case (***Heat Input Equation Code***)
= "F-15":
If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for HI*** is not null AND ***Valid FC Factor Exists*** == true AND ***FLOW Calculated Adjusted Value*** is not null)

HI Calculated Adjusted Value = (***FLOW Calculated Adjusted Value*** * ***Calculated Diluent for Heat Input***) / (***Current Hourly Op Record***.FcFactor * 100.0), and round the result to 1 decimal place.
else
return result A

= "F-16":
If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for HI*** is not null AND ***Valid FC Factor Exists*** == true AND ***FLOW Calculated Adjusted Value*** is not null AND ***Calculated Moisture for HI*** is not null)

HI Calculated Adjusted Value = [***FLOW Calculated Adjusted Value*** * (100 - ***Calculated Moisture for HI***) * ***Calculated Diluent for HI***] / (10,000 * ***Current Hourly Op Record***.FcFactor), and round the result to 1 decimal place.

else

return result A

= "F-17":

If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for HI*** is not null AND ***Valid FD Factor Exists*** == true AND ***FLOW Calculated Adjusted Value*** is not null AND ***Calculated Moisture for HI*** is not null)

HI Calculated Adjusted Value = [***FLOW Calculated Adjusted Value*** * (1 / ***Current Hourly Op Record***.FdFactor)* [0.209 * (100 - ***Calculated Moisture for HI***) - ***Calculated Diluent for HI***] / 20.9)], and round the result to 1 decimal place.

else

return result A

= "F-18":

If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for HI*** is not null AND ***Valid FD Factor Exists*** == true AND ***FLOW Calculated Adjusted Value*** is not null AND ***Calculated Moisture for HI*** is not null)

HI Calculated Adjusted Value = (***FLOW Calculated Adjusted Value*** *[100 - ***Calculated Moisture for HI***] * [20.9 - ***Calculated Diluent for HI***]) / (2090 * ***Current Hourly Op Record***.FdFactor), and round the result to 1 decimal place.

else

return result A

= All Other Equation Codes:

return result A

if no result

if (***HI Calculated Adjusted Value*** is less than 1 AND ***Legacy Data Evaluation*** == false)

HI Calculated Adjusted Value = 1

Apportionment Calc HI Array at this Location = ***HI Calculated Adjusted Value***

if (***MP Stack Config for Hourly Checks*** == "MS")

Config HeatInput Accumulator = ***Config HeatInput Accumulator*** + ***HI Calculated Adjusted Value***

if result A

Apportionment Calc HI Array at this Location = -1

Config HeatInputTimesOpTime Accumulator = -1

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)

Rpt Period HI Calculated Accumulator Array for this location = -1

if (***MP Stack Config for Hourly Checks*** == "MS")

Config HeatInput Accumulator = -1

else if (***Heat Input App D Method Active for Hour*** == true)

if (***HI App D Accumulator*** >= 0)

Total Heat Input from Fuel Flow = ***HI App D Accumulator***

if (***HI App D Accumulator*** >= 0 AND ***Current Hourly Op Record***.OperatingTime is greater than 0 and less than or equal to 1)

Apportionment Calc HI Array at this Location = ***HI App D Accumulator*** / ***Current Hourly Op Record***.OperatingTime, rounded to one decimal place.

if (***Heat Input Method Code*** == "AD")
HI Calculated Adjusted Value = ***Apportionment Calc HI Array*** at this Location
App E Calc HI = ***HI Calculated Adjusted Value***
else
for each location in the configuration where ***Apportionment HI Method Array*** in set {CALC, ADCALC}
set ***Apportionment HI Method Array*** for this location to "NOCALC"
else
Apportionment Calc HI Array at this Location = -1
Config HeatInputTimesOpTime Accumulator = -1
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period HI Calculated Accumulator Array for this location = -1
return result A

else if (***Heat Input Method Code*** NOT in set {ADCALC, CALC})
HI Calculated Adjusted Value = ***Current DHV Calculated Adjusted Value***
Apportionment Calc HI Array at this Location = ***HI Calculated Adjusted Value***

If (***HI Calculated Adjusted Value*** is not null)

If (***Current Hourly Op Record***.OperatingTime is between 0 and 1 inclusive)

Heat Input Total Calculated Value = ***HI Calculated Adjusted Value*** * ***Current Hourly Op Record***.OperatingTime.

if (***Config HeatInputTimesOpTime Accumulator*** >= 0)
Config HeatInputTimesOpTime Accumulator = ***Config HeatInputTimesOpTime Accumulator*** + *Heat Input Total Calculated Value*

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
if (***Rpt Period HI Calculated Accumulator Array*** for this location is not null)
if (***Rpt Period HI Calculated Accumulator Array*** for this location >= 0)
Rpt Period HI Calculated Accumulator Array for this location = ***Rpt Period HI Calculated Accumulator Array*** for this location + *Heat Input Total Calculated Value*
else
Rpt Period HI Calculated Accumulator Array for this location = *Heat Input Total Calculated Value*

if (***Current Month*** is April)
if (***April HI Calculated Accumulator Array*** for this location is not null)
April HI Calculated Accumulator Array for this location = ***April HI Calculated Accumulator Array*** for this location + *Heat Input Total Calculated Value*
else
April HI Calculated Accumulator Array for this location = *Heat Input Total Calculated Value*

else
Config HeatInputTimesOpTime Accumulator = -1
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period HI Calculated Accumulator Array for this location = -1

If (***Derived Hourly Adjusted Value Status*** == true)

If (***Heat Input Method Code*** in set {CEM, AD})

if (***HI Calculated Adjusted Value*** is equal to 1 AND ***Current DHV Record***.AdjustedHourlyValue is less than 1 AND ***Current DHV Record***.MODCCode is not equal to "26" and ***Legacy Data Evaluation*** == false)
return result C

else

Heat Input Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "HI" AND
UOM = "MMBTUHR"

if (ABS(***Current DHV Record***.AdjustedHourlyValue - ***HI Calculated Adjusted Value***) > *Heat Input Tolerance*)
return result B

else if (***Apportionment Calc HI Array*** at this Location is greater than or equal to 0) \\ ADCALC

If (***Current Hourly Op Record*****.**OperatingTime is between 0 and 1 inclusive)

Heat Input Total Calculated Value = ***Apportionment Calc HI Array*** at this Location * ***Current Hourly Op Record*****.**OperatingTime.

if (***Config HeatInputTimesOpTime Accumulator*** >= 0)

Config HeatInputTimesOpTime Accumulator = ***Config HeatInputTimesOpTime Accumulator*** + *Heat Input Total Calculated Value*

else
Config HeatInputTimesOpTime Accumulator = -1

else if (***Heat Input Method Code*** not in set {ADCALC, CALC})
Apportionment Calc HI Array at this Location = -1
Config HeatInputTimesOpTime Accumulator = -1
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period HI Calculated Accumulator Array for this location = -1

Results:

Result	Response	Severity
A	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated due to errors listed above.	Informational Message
B	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1
C	You reported in AdjustedHourlyValue of less than 1 in the DHV record for [param]. You must report a minimum heat input of 1 and a MODCCode of "26".	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Heat Input Calculation Verification

Check Code: HOURCV-9

Check Name: Calculate SO2 Mass Emissions

Related Former Checks:

Applicability: CEM Check

Description: Based on values from SO2 Monitor Hourly and Stack Flow Monitor Hourly, plus moisture values if applicable and the current equation code, the SO2 Mass emissions rate is calculated

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

if (***Derived Hourly Adjusted Value Status*** == true AND ***Current Hourly Op Record*****.**OperatingTime is between 0 and 1 (inclusive))

SO2 Total Reported Value = ***Current DHV Record***.AdjustedHourlyValue * ***Current Hourly Op Record*****.**OperatingTime.
if (***Rpt Period SO2 Mass Reported Accumulator Array*** for this location is not null)
if (***Rpt Period SO2 Mass Reported Accumulator Array*** for this location >= 0)
Rpt Period SO2 Mass Reported Accumulator Array for this location = ***Rpt Period SO2 Mass Reported Accumulator Array*** for this location + *SO2 Total Reported Value*
else
Rpt Period SO2 Mass Reported Accumulator Array for this location = *SO2 Total Reported Value*
else
Rpt Period SO2 Mass Reported Accumulator Array for this location = -1

If (***SO2 CEM Method Active for Hour*** == true)

if (***SO2 Equation Code*** == "F-1")

If (***Current DHV Record Valid*** == true AND ***SO2C Calculated Adjusted Value*** is not null AND ***FLOW Calculated Adjusted Value*** is not null)

SO2 Calculated Adjusted Value = 0.000000166 * ***SO2C Calculated Adjusted Value * FLOW Calculated Adjusted Value,*** ROUNDED to one decimal place.
else
Rpt Period SO2 Mass Calculated Accumulator Array for this location = -1
return result A

else if (***SO2 Equation Code*** == "F-2")

If (***Current DHV Record Valid*** == true AND ***SO2C Calculated Adjusted Value*** is not null AND ***FLOW Calculated Adjusted Value*** is not null AND ***Calculated Moisture for SO2*** is not null)

SO2 Calculated Adjusted Value = 0.000000166 * ***SO2C Calculated Adjusted Value * FLOW Calculated Adjusted Value*** * (100.0 - ***Calculated Moisture for SO2***) / 100.0**,** ROUNDED to one decimal place.
else
Rpt Period SO2 Mass Calculated Accumulator Array for this location = -1
return result A
else
Rpt Period SO2 Mass Calculated Accumulator Array for this location = -1
return result A

else if (***SO2 F23 Method Active for Hour*** == true)

If (***Current DHV Record Valid*** == true AND ***F23 Default Value*** is not null AND ***HI Calculated Adjusted Value*** is not null)

SO2 Calculated Adjusted Value = ***F23 Default Value * HI Calculated Adjusted Value***, rounded to one decimal

place.

else

Rpt Period SO2 Mass Calculated Accumulator Array for this location = -1
return result A

else if (***SO2 App D Method Active for Hour*** == true)

if (***SO2 App D Accumulator*** >= 0 AND ***Current Hourly Op Record***.OperatingTime is between 0 and 1 (inclusive))
SO2 Calculated Adjusted Value = ***SO2 App D Accumulator*** / ***Current Hourly Op Record***.OperatingTime.

If (***Hourly Fuel Flow Count For Gas*** is greater than 0)
Round ***SO2 Calculated Adjusted Value*** to four decimal places.
else
Round ***SO2 Calculated Adjusted Value*** to one decimal place.

else
Rpt Period SO2 Mass Calculated Accumulator Array for this location = -1
return result A
else
SO2 Calculated Adjusted Value = ***Current DHV Calculated Adjusted Value***

If (***SO2 Calculated Adjusted Value*** is not null)

If (***Current Hourly Op Record***.OperatingTime is between 0 and 1 inclusive)

SO2 Total Calculated Value = ***SO2 Calculated Adjusted Value * Current Hourly Op Record***.OperatingTime

if (***Rpt Period SO2 Mass Calculated Accumulator Array*** for this location is not null)
if (***Rpt Period SO2 Mass Calculated Accumulator Array*** for this location >= 0)
Rpt Period SO2 Mass Calculated Accumulator Array for this location = ***Rpt Period SO2 Mass Calculated Accumulator Array*** for this location + *SO2 Total Calculated Value*
else
Rpt Period SO2 Mass Calculated Accumulator Array for this location = *SO2 Total Calculated Value*
else
Rpt Period SO2 Mass Calculated Accumulator Array for this location = -1

If (***Derived Hourly Adjusted Value Status*** == true)

SO2 Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "SO2" AND
UOM = "LBHR"

if (ABS(***Current DHV Record***.AdjustedHourlyValue - ***SO2 Calculated Adjusted Value***) > *SO2 Tolerance*)
return result B
else
Rpt Period SO2 Mass Calculated Accumulator Array for this location = -1

Results:

Result	Response	Severity
A	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated due to errors listed above.	Informational Message
B	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ SO2 Calculation Verification

Check Code: HOURCV-12

Check Name: Determine Diluent Cap, Moisture, and NOXC for NOx Rate Calculation Verification

Related Former Checks:

Applicability: CEM Check

Description: For each possible formula, ensure that all required components are available. Note that no responses are output, because each missing part should have generated an earlier error message.

Specifications:

If (***Current NOx Rate Method Code*** == "CEM" AND ***Current DHV Record***.ModcCode in set {01, 02, 03, 04, 14, 22, 53, 54})

if (***NOx Conc Monitor Hourly Count*** == 1 AND ***Current NOx Conc Monitor Hourly Record***.UnadjustedHourlyValue is not null)
NOx Conc for NOx Rate Calculation = Current NOx Conc Monitor Hourly Record.UnadjustedHourlyValue
else
NOx Conc for NOx Rate Calculation = null

if (***NOx Rate Equation Code*** in set {19-3, 19-3D, 19-4, 19-5, 19-8, 19-9})
If (***H2O Method Code*** == "MWD" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
Calculated Moisture for NOXR = H2O DHV Calculated Adjusted Value
else if (***H2O Method Code*** in set {MMS, MTB} AND ***H2O Monitor Hourly Checks Needed*** == true AND ***H2O MHV Calculated Adjusted Value*** is not null)
Calculated Moisture for NOXR = H2O MHV Calculated Adjusted Value
else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
Calculated Moisture for NOXR = H2O DHV Calculated Adjusted Value
else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == false AND ***H2O Default Value*** is not null)
Calculated Moisture for NOXR = H2O Default Value

if (***NOx Rate Equation Code*** in set {19-3D, 19-5D} OR ***Current DHV Record***.ModcCode == 14)

If (***NOx Rate Equation Code*** in set {F-5, 19-1, 19-2, 19-3, 19-3D, 19-4, 19-5, 19-5D})

O2 Monitor Default Matches = count of # active MonitoringDefaultData records for location where
MonitoringDefaultData.ParameterCode = "O2X" AND MonitoringDefaultData.DefaultPurposeCode = "DC" AND MonitoringDefaultData.FuelCode = "NFS"

if *O2 Monitor Default Matches* > 1
return result A
else if *O2 Monitor Default Matches* = 0
return result B
else
O2 Monitor Default Record = the single matched record
if (*O2 Monitor Default Record*.DefaultValue is NULL OR *O2 Monitor Default Record*.DefaultValue <= 0)
return result C
else
Calculated Diluent for NOXR = *O2 Monitor Default Record*.DefaultValue

else if (***NOx Rate Equation Code*** in set {F-6, 19-6, 19-7, 19-8, 19-9})

CO2 Monitor Default Matches = count of # active MonitoringDefaultData records for location where
MonitoringDefaultData.ParameterCode = "CO2N" AND MonitoringDefaultData.DefaultPurposeCode = "DC" AND MonitoringDefaultData.FuelCode = "NFS"

if *CO2 Monitor Default Matches* > 1

return result D
else if *CO2 Monitor Default Matches* = 0
return result E
else
CO2 Monitor Default Record = the single matched record
if (*CO2 Monitor Default Record*.DefaultValue is NULL OR *CO2 Monitor Default Record*.DefaultValue <= 0)
return result F
else
Calculated Diluent for NOx Rate = *CO2 Monitor Default Record*.DefaultValue

else

If (***NOx Rate Equation Code*** in set {F-5, 19-1, 19-4} AND ***O2 Dry Checks Needed for NOx Rate Calc*** == true)
Calculated Diluent for NOx Rate = ***O2 Dry Calculated Adjusted Value***

else if (***NOx Rate Equation Code*** in set {19-2, 19-3, 19-5} AND ***O2 Wet Checks Needed for NOx Rate Calc*** == true)
Calculated Diluent for NOx Rate = ***O2 Wet Calculated Adjusted Value***

else if (***NOx Rate Equation Code*** in set {F-6, 19-6, 19-7, 19-8, 19-9} AND ***CO2 Conc Monitor Hourly Checks Needed*** == true)
Calculated Diluent for NOx Rate = ***CO2C MHV Calculated Adjusted Value***

Results:

Result	Response	Severity
A	You reported more than one diluent cap default record for O2X in your monitoring plan that was active during current hour.	Critical Error Level 1
B	You did not report a default record for O2X in your monitoring plan that was active during current hour.	Critical Error Level 1
C	The DefaultValue reported in the active Default record for O2X in your monitoring plan is invalid. The value must be greater than 0.	Critical Error Level 1
D	You reported more than one diluent cap default record for CO2N in your monitoring plan that was active during the current hour.	Critical Error Level 1
E	You did not report an active CO2N diluent cap default record in your monitoring plan for the hour.	Critical Error Level 1
F	The DefaultValue reported in the active Default record for CO2N in your monitoring plan is invalid. The value must be greater than 0.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification

Check Code: HOURCV-13

Check Name: Calculate Unadjusted NOx Emissions Rate

Related Former Checks:

Applicability: CEM Check

Description: Based on equation code in NOx Emission Rate record and reported values, calculate the NOx Emissions Rate

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

NOXR Calculated Unadjusted Value = null
Derived Hourly Unadjusted Calculation Status = false

If (***Current NOx Rate Method Code*** == "CEM" AND ***Current DHV Record.***ModcCode in set {01, 02, 03, 14, 22, 53})
If (***Current DHV Record***.SystemTypeCode == "NOX")
RATA Status Required = true

case (***NOx Rate Equation Code***)

"19-1" or "F-5":
If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for NOXR*** is not null AND ***NOX Conc for NOx Rate Calc*** is not null AND ***Valid FD Factor Exists*** == true)

if (***Calculated Diluent for NOXR*** == 20.9)
return result A
else
NOXR Calculated Unadjusted Value = 0.0000001194 * ***NOX Conc for NOx Rate Calc*** * ***Current Hourly Op Record***.FdFactor * [20.9 / (20.9 - ***Calculated Diluent for NOXR***)], rounded to 3 decimal places.

else
return result B

"19-2":
Moisture Fraction = null
BWA Default Record Count = count active MonitoringDefaultData Records for the location where ParameterCd = 'BWA'

If (*BWA Default Record Count* == 0)
Moisture Fraction = 0.027
else If (*BWA Default Record Count* == 1 AND MonitorDefaultData.DefaultValue > 0 AND MonitorDefaultData.DefaultValue < 1)
Moisture Fraction = MonitorDefaultData.DefaultValue
else
return result D

If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for NOXR*** is not null AND ***NOX Conc for NOx Rate Calc*** is not null AND ***Valid FW Factor Exists*** == true AND *Moisture Fraction* is not null)

if (***Calculated Diluent for NOXR*** == 20.9 * (1 - *Moisture Fraction*))
return result A
else

NOXR Calculated Unadjusted Value = 0.0000001194 * ***NOX Conc for NOx Rate Calc*** * ***Current Hourly Op Record***.FwFactor * [20.9 / (20.9 *(1 - *Moisture Fraction*) - ***Calculated Diluent for NOXR***)], rounded to 3 decimal places.
else

return result B

"19-3":

If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for NOXR*** is not null AND ***NOX Conc for NOx Rate Calc*** is not null AND ***Valid FD Factor Exists*** == true AND ***Calculated Moisture for NOXR*** is not null)

if (***Calculated Diluent for NOXR*** == 20.9*(100 - ***Calculated Moisture for NOXR***) / 100)
return result A
else
denom = (20.9*(100 - ***Calculated Moisture for NOXR***)/100 - ***Calculated Diluent for NOXR***)
NOXR Calculated Unadjusted Value = 0.0000001194 * ***NOX Conc for NOx Rate Calc*** * ***Current Hourly Op Record***.FdFactor * [20.9 /denom], rounded to 3 decimal places.

else
return result B

"19-3D":

If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for NOXR*** is not null AND ***NOX Conc for NOx Rate Calc*** is not null AND ***Valid FD Factor Exists*** == true AND ***Calculated Moisture for NOXR*** is not null)

h2oFactor = (100 - ***Calculated Moisture for NOXR***) / 100.0
denomTerm = (20.9 * *h2oFactor*) - (***Calculated Diluent for NOXR*** * *h2oFactor*)
if (*denomTerm* == *0*)
return result A
else
NOXR Calculated Unadjusted Value = 0.0000001194 * ***NOX Conc for NOx Rate Calc*** * ***Current Hourly Op Record***.FdFactor * 20.9 / *denomTerm*, rounded to 3 decimal places.

else
return result B

"19-4":

If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for NOXR*** is not null AND ***NOX Conc for NOx Rate Calc*** is not null AND ***Valid FD Factor Exists*** == true AND ***Calculated Moisture for NOXR*** is not null)

if (***Calculated Diluent for NOXR*** == 20.9 OR ***Calculated Moisture for NOXR*** == 100)
return result A
else

NOXR Calculated Unadjusted Value = 0.0000001194 * ***NOX Conc for NOx Rate Calc*** * ***Current Hourly Op Record***.FdFactor / [(100 - ***Calculated Moisture for NOXR***) / 100.0] * (20.9 / (20.9 - ***Calculated Diluent for NOXR***)), rounded to 3 decimal places.

else
return result B

"19-5":

If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for NOXR*** is not null AND ***NOX Conc for NOx Rate Calc*** is not null AND ***Valid FD Factor Exists*** == true AND ***Calculated Moisture for NOXR*** is not null)

if (***Calculated Moisture for NOXR*** == 100)
return result A

else
H2OTerm = (100 - ***Calculated Moisture for NOXR***) / 100.0
denom = 20.9 - ***Calculated Diluent for NOXR*** / H2OTerm

if (*denom* == 0)
return result A

else
NOXR Calculated Unadjusted Value = 0.0000001194 * ***NOX Conc for NOx Rate Calc*** * ***Current Hourly Op Record***.FdFactor / *denom*, rounded to 3 decimal places.

else
return result B

"19-5D":
If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for NOXR*** is not null AND ***NOX Conc for NOx Rate Calc*** is not null AND ***Valid FD Factor Exists*** == true)

if (***Calculated Diluent for NOXR*** == 20.9)
return result A
else
NOXR Calculated Unadjusted Value = 0.0000001194 * ***NOX Conc for NOx Rate Calc*** * ***Current Hourly Op Record***.FdFactor * 20.9/ (20.9 - ***Calculated Diluent for NOXR***), rounded to 3 decimal places.

else
return result B

"19-6" or "19-7" or "F-6":
If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for NOXR*** is not null AND ***NOX Conc for NOx Rate Calc*** is not null AND ***Valid FC Factor Exists*** == true)

if (***Calculated Diluent for NOXR*** == 0.0)
return result A
else
NOXR Calculated Unadjusted Value = 0.0000001194 * ***NOX Conc for NOx Rate Calc*** * ***Current Hourly Op Record***.FcFactor * 100.0 / ***Calculated Diluent for NOXR*** , rounded to 3 decimal places.

else
return result B

"19-8":
If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for NOXR*** is not null AND ***NOX Conc for NOx Rate Calc*** is not null AND ***Valid FC Factor Exists*** == true AND ***Calculated Moisture for NOXR*** is not null)

if (***Calculated Diluent for NOXR*** == 0.0 OR ***Calculated Moisture for NOXR*** == 100)
return result A
else
NOXR Calculated Unadjusted Value = 0.0000001194 * ***NOX Conc for NOx Rate Calc*** * ***Current Hourly Op Record***.FcFactor / [(100 - ***Calculated Moisture for NOXR***) / 100.0] * (100.0 / ***Calculated Diluent for NOXR***), rounded to 3 decimal places.

else
return result B

"19-9":
If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for NOXR*** is not null AND ***NOX Conc for NOx Rate Calc*** is not null AND ***Valid FC Factor Exists*** == true AND ***Calculated Moisture for NOXR*** is not null)

if (***Calculated Diluent for NOXR*** == 0.0)
return result A

else

H2OTerm = (100 - ***Calculated Moisture for NOXR***) / 100.0
CO2Term = 100.0 / ***Calculated Diluent for NOXR***
NOXR Calculated Unadjusted Value = 0.0000001194 * ***NOX Conc for NOx Rate Calc*** * ***Current Hourly Op Record***.FcFactor * *H2OTerm* * *CO2Term*, rounded to 3 decimal places.

else

return result B

If (***Derived Hourly Unadjusted Value Status*** == true AND ***NOXR Calculated Unadjusted Value*** is not null)

Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "NOXR" AND
UOM = "LBMMBTU"

if (ABS(***Current DHV Record***.UnadjustedHourlyValue - ***NOXR Calculated Unadjusted Value***) > *Tolerance*)
return result C
else
Derived Hourly Unadjusted Calculation Status = true

else if (***Current NOx Rate Method Code*** == "PEM" AND ***Current DHV Record.***ModcCode in set {01, 02, 03})
If (***Current DHV Record***.SystemTypeCode == "NOXP")
RATA Status Required = true

If ***Current DHV Record***.UnadjustedHourlyValue >= 0
NOXR Calculated Unadjusted Value = ***Current DHV Record***.UnadjustedHourlyValue
Derived Hourly Unadjusted Calculation Status = true

else if (***Current NOx Rate Method Code*** == "AE")

If (***App E Constant Fuel Mix*** == true)
NOXR Calculated Adjusted Value = ***App E Calculated NOx Rate for Source***

else
NOXR Calculated Adjusted Value = ***Current DHV Calculated Adjusted Value***

if (***Current NOx Rate Method Code*** in set {CEM, PEM} AND ***Current DHV Record.***ModcCode == 21)
If (***Current DHV Record***.SystemTypeCode in set {NOX, NOXP})
RATA Status Required = true

Results:

Result	Response	Severity
A	The NOx emission rate could not be recalculated, because the diluent and/or moisture value would result in division by zero. You should report an MODC of 14 indicating the use of a diluent cap to prevent this.	Critical Error Level 1
B	The UnadjustedHourlyValue in the DHV record for [param] could not be recalculated due to errors listed above.	Informational Message
C	The UnadjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1
D	You did not report a single valid MonitorDefault record for ParameterCode BWA for the hour.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification

Check Code: HOURCV-15

Check Name: Determine Moisture for NOx Mass Calculation Verification

Related Former Checks:

Applicability: CEM Check

Description: Verifies that all elements are present to support the equation code indicated by the current NOx Mass Rate record

Specifications:

If (***NOx Mass Monitor Method Code*** begins with "CEM")
If (***NOx Mass Equation Code*** == "F-26B")
If (***H2O Method Code*** == "MWD" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
Calculated Moisture for NOX = H2O DHV Calculated Adjusted Value
else if (***H2O Method Code*** in set {MMS, MTB} AND ***H2O Monitor Hourly Checks Needed*** == true AND ***H2O MHV Calculated Adjusted Value*** is not null)
Calculated Moisture for NOX = H2O MHV Calculated Adjusted Value
else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
Calculated Moisture for NOX = H2O DHV Calculated Adjusted Value
else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == false AND ***H2O Default Value*** is not null)
Calculated Moisture for NOX = H2O Default Value

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Calculation Verification

Check Code: HOURCV-16

Check Name: Calculate NOx Mass Emissions

Related Former Checks:

Applicability: CEM Check

Description: Based on Formula Code and all reported values, the NOx Mass Calculation is verified

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

if (***Derived Hourly Adjusted Value Status*** == true AND ***Current Hourly Op Record***.OperatingTime is between 0 and 1 (inclusive))

NOX Mass Total Reported Value = ***Current DHV Record***.AdjustedHourlyValue * ***Current Hourly Op Record***.OperatingTime.
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
if (***Rpt Period NOX Mass Reported Accumulator Array*** for this location is not null)
if (***Rpt Period NOX Mass Reported Accumulator Array*** for this location >= 0)
Rpt Period NOX Mass Reported Accumulator Array for this location = ***Rpt Period NOX Mass Reported Accumulator Array*** for this location + *NOX Mass Total Reported Value*
else
Rpt Period NOX Mass Reported Accumulator Array for this location = *NOX Mass Total Reported Value*

else
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period NOX Mass Reported Accumulator Array for this location = -1

If (***NOx Mass Monitor Method Code*** in set {CEM, NOXR, CEMNOXR})

if (***NOx Mass Equation Code*** == "F-26A")

If (***Current DHV Record Valid*** == true AND ***NOXC Calculated Adjusted Value*** is not null AND ***FLOW Calculated Adjusted Value*** is not null)

NOX Calculated Adjusted Value = 0.0000001194 * ***NOXC Calculated Adjusted Value*** * ***Stack Flow Calculated Adjusted Value,*** ROUNDED to one decimal place.
else
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period NOX Mass Calculated Accumulator Array for this location = -1
return result A

else if (***NOX Mass Equation Code*** == "F-26B")

If (***Current DHV Record Valid*** == true AND ***NOXC Calculated Adjusted Value*** is not null AND ***FLOW Calculated Adjusted Value*** is not null AND ***Calculated Moisture for NOX*** is not null)

NOX Calculated Adjusted Value = 0.0000001194 * ***NOXC Calculated Adjusted Value*** * ***FLOW Calculated Adjusted Value*** * (100.0 - ***Calculated Moisture for NOX***) / 100.0, ROUNDED to one decimal place.
else
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period NOX Mass Calculated Accumulator Array for this location = -1
return result A

else if (***NOX Mass Equation Code*** == "F-24A")

if (***Heat Input Method Code*** NOT in set {CALC, ADCALC})

If (***Current DHV Record Valid*** == true AND ***NOXR Calculated Adjusted Value*** is not null

If (***HI Calculated Adjusted Value*** is not null)

NOX Calculated Adjusted Value = ***NOXR Calculated Adjusted Value*** * ***HI Calculated Adjusted Value,*** ROUNDED to one decimal place.

else

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)

Rpt Period NOX Mass Calculated Accumulator Array for this location = -1

return result A

else

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)

Rpt Period NOX Mass Calculated Accumulator Array for this location = -1

return result A

else

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)

Rpt Period NOX Mass Calculated Accumulator Array for this location = -1

return result A

else

NOX Calculated Adjusted Value = ***Current DHV Calculated Adjusted Value***

If (***NOX Calculated Adjusted Value*** is not null)

If (***Current Hourly Op Record.***OperatingTime is between 0 and 1 inclusive)

NOx Mass Total Calculated Value = ***NOX Calculated Adjusted Value*** * ***Current Hourly Op Record.***OperatingTime.

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)

if (***Rpt Period NOX Mass Calculated Accumulator Array*** for this location is not null)

if (***Rpt Period NOX Mass Calculated Accumulator Array*** for this location >= 0)

Rpt Period NOX Mass Calculated Accumulator Array for this location = ***Rpt Period NOX Mass Calculated Accumulator Array*** for this location + *NOX Mass Total Calculated Value*

else

Rpt Period NOX Mass Calculated Accumulator Array for this location = *NOX Mass Total Calculated Value*

if (***Current Month*** is April)

if (***April NOX Mass Calculated Accumulator Array*** for this location is not null)

April NOX Mass Calculated Accumulator Array for this location = ***April NOX Mass Calculated Accumulator Array*** for this location + *NOX Mass Total Calculated Value*

else

April NOX Mass Calculated Accumulator Array for this location = *NOX Mass Total Calculated Value*

else

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)

Rpt Period NOX Mass Calculated Accumulator Array for this location = -1

If (***Derived Hourly Adjusted Value Status*** == true AND ***Current DHV Method*** in set {CEM, NOXR, CEMNOXR})

NOX Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where

Parameter = "NOX" AND

UOM = "LBHR"

if (ABS(***Current DHV Record***.AdjustedHourlyValue - ***NOX Calculated Adjusted Value***) > *NOX Tolerance*)

If (***Legacy Data Evaluation*** == false)

return result B

else if (***Current Hourly Op Data***.OpTime is greater than 0 and less than or equal to 1)

if (ABS(***Current DHV Record***.AdjustedHourlyValue - ***NOX Calculated Adjusted Value***) > *NOX Tolerance* / ***Current Hourly Op Record***.OperatingTime)
return result B

else if (***NOX Mass Equation Code*** <> "F-24A" OR ***Current DHV Record Valid*** == false OR ***NOXR Calculated Adjusted Value*** is null OR ***Heat Input Method Code*** NOT in set {CALC, ADCALC})
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period NOX Mass Calculated Accumulator Array for this location = -1

Results:

Result	Response	Severity
A	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated due to errors listed above.	Informational Message
B	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ NOx Mass Rate Calculation Verification

Check Code: HOURCV-18

Check Name: Determine Diluent Cap and Moisture for CO2 Mass Calculation Verification

Related Former Checks:

Applicability: CEM Check

Description: For each possible formula, ensure that all required components are available. Note that no responses are output, because each missing part should have generated an earlier error message.

Specifications:

If (***CO2 Method Code*** == "CEM")
    if (***CO2 Mass Equation Code*** == "F-2")
        If (***H2O Method Code*** == "MWD" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
            Calculated Moisture for CO2 = H2O DHV Calculated Adjusted Value
        else if (***H2O Method Code*** in set {MMS, MTB} AND ***H2O Monitor Hourly Checks Needed*** == true AND ***H2O MHV Calculated Adjusted Value*** is not null)
            Calculated Moisture for CO2 = H2O MHV Calculated Adjusted Value
        else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
            Calculated Moisture for CO2 = H2O DHV Calculated Adjusted Value
        else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == false AND ***H2O Default Value*** is not null)
            Calculated Moisture for CO2 = H2O Default Value

    if (***Use CO2 Diluent Cap for CO2 Mass Calc*** == true)
        CO2N Count = # of active MonitoringDefault records for location where
            ParameterCode = 'CO2N' AND DefaultPurposeCode = 'DC' AND
            FuelCode = 'NFS'
        if (*CO2N Count* > 1)
            return result A
        else if (*CO2N Count* == 0)
            return result B
        else if MonitoringDefault.DefaultValue <= 0
            return result C
        else
            Calculated Diluent for CO2 = MonitoringDefault.DefaultValue

    else if (***CO2 Conc Derived Checks Needed*** == true)
        Calculated Diluent for CO2 = CO2C DHV Calculated Adjusted Value

    else if (***CO2 Conc Checks Needed for CO2 Mass*** == true)
        if (***Current CO2 Conc Missing Data Monitor Hourly Record*** is not null)
            Calculated Diluent for CO2 = CO2C SD Calculated Adjusted Value
        else
            Calculated Diluent for CO2 = CO2C MHV Calculated Adjusted Value

Results:

Result	Response	Severity
A	You reported more than one diluent cap default record for CO2N in your monitoring plan that was active during the current hour.	Critical Error Level 1
B	A DHV record indicates use of a diluent cap to calculate CO2, but you did not report an active CO2N default record in your monitoring plan for the hour. Please note that the use of the diluent cap to calculate CO2 is only applicable to legacy data.	Critical Error Level 1
C	The DefaultValue reported in the active Default record for CO2N in your monitoring plan is invalid. The value must be greater than 0.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2 Mass Rate Calculation Verification

Check Code: HOURCV-19

Check Name: Calculate CO2 Mass Emissions

Related Former Checks:

Applicability: CEM Check

Description: Based on values from CO2 Monitor Hourly and Stack Flow Monitor Hourly, plus moisture values if applicable and the current equation code, the CO2 Mass emissions rate is calculated

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

if (***Derived Hourly Adjusted Value Status*** == true AND ***Current Hourly Op Record*****.**OperatingTime is between 0 and 1 (inclusive))

CO2 Total Reported Value = ***Current DHV Record***.AdjustedHourlyValue **** Current Hourly Op Record*****.**OperatingTime.

if (***Rpt Period CO2 Mass Reported Accumulator Array*** for this location is not null)
if (***Rpt Period CO2 Mass Reported Accumulator Array*** for this location >= 0)
Rpt Period CO2 Mass Reported Accumulator Array for this location = ***Rpt Period CO2 Mass Reported Accumulator Array*** for this location + *CO2 Total Reported Value*
else
Rpt Period CO2 Mass Reported Accumulator Array for this location = *CO2 Total Reported Value*
else
Rpt Period CO2 Mass Reported Accumulator Array for this location = -1

If (***CO2 Method Code*** == "CEM")

if (***CO2 Mass Equation Code*** == "F-11")

If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for CO2*** is not null AND ***FLOW Calculated Adjusted Value*** is not null)

CO2 Calculated Adjusted Value = 0.00000057 **** Calculated Diluent for CO2 * FLOW Calculated Adjusted Value,*** ROUNDED to one decimal place.
else
Rpt Period CO2 Mass Calculated Accumulator Array for this location = -1
return result A

else if (***CO2 Mass Equation Code*** == "F-2")

If (***Current DHV Record Valid*** == true AND ***Calculated Diluent for CO2*** is not null AND ***FLOW Calculated Adjusted Value*** is not null AND ***Calculated Moisture for CO2*** is not null)

CO2 Calculated Adjusted Value = 0.00000057 **** Calculated Diluent for CO2 * FLOW Calculated Adjusted Value**** [(100.0 - ***Calculated Moisture for CO2***) / 100.0]**,** ROUNDED to one decimal place.
else
Rpt Period CO2 Mass Calculated Accumulator Array for this location = -1
return result A
else
Rpt Period CO2 Mass Calculated Accumulator Array for this location = -1
return result A

else if (***CO2 App D Method Active for Hour*** == true)

if (***CO2 App D Accumulator*** >= 0 AND ***Current Hourly Op Record*****.**OperatingTime is between 0 and 1 (inclusive))
CO2 Calculated Adjusted Value = ***CO2 App D Accumulator*** / ***Current Hourly Op Record***.OperatingTime, rounded to one decimal place.

else
if (***Legacy Data Evaluation*** == false)
Rpt Period CO2 Mass Calculated Accumulator Array for this location = -1
return result A
else
Rpt Period CO2 Mass Calculated Accumulator Array for this location = -2
else
CO2 Calculated Adjusted Value = ***Current DHV Calculated Adjusted Value***

If (***CO2 Calculated Adjusted Value*** is not null)

If (***Current Hourly Op Record***.OperatingTime is between 0 and 1 inclusive)

CO2 Total Calculated Value = ***CO2 Calculated Adjusted Value*** * ***Current Hourly Op Record***.OperatingTime.

if (***Rpt Period CO2 Mass Calculated Accumulator Array*** for this location is not null)
if (***Rpt Period CO2 Mass Calculated Accumulator Array*** for this location >= 0)
Rpt Period CO2 Mass Calculated Accumulator Array for this location = ***Rpt Period CO2 Mass Calculated Accumulator Array*** for this location + *CO2 Total Calculated Value*
else
Rpt Period CO2 Mass Calculated Accumulator Array for this location = *CO2 Total Calculated Value*
else
Rpt Period CO2 Mass Calculated Accumulator Array for this location = -1

If (***Derived Hourly Adjusted Value Status*** == true)

CO2 Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "CO2" AND
UOM = "TNHR"

if (ABS(***Current DHV Record***.AdjustedHourlyValue - ***CO2 Calculated Adjusted Value***) > *CO2 Tolerance*)
return result B
else
if (***Rpt Period CO2 Mass Calculated Accumulator Array*** for this location <> -2)
Rpt Period CO2 Mass Calculated Accumulator Array for this location = -1

Results:

Result	Response	Severity
A	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated due to errors listed above.	Informational Message
B	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2 Mass Rate Calculation Verification

Check Code: HOURCV-25

Check Name: Determine BAF Value for NOx Emission Rate System

Related Former Checks:

Applicability: CEM Check

Description: Retrieves and sets as an output parameter the Bias Adjustment factor for the NOX Rate Monitoring System

Specifications:

Current NOX System BAF = null

If (***Current NOx System Status*** == true AND ***NOXR Calculated Unadjusted Value*** is not null AND ***Current NOx Rate Method Code*** in set {CEM, PEM}) AND ***Current DHV Record.***ModcCode in set {01, 02, 03, 14, 22, 53})

If (***RATA Status BAF*** is not null)
Current NOX System BAF = RATA Status BAF
else
return result A

Results:

Result	Response	Severity
A	The BAF for [ParamCode] MonitoringSystemID [ID] cannot be determined, because the prior RATA had critical errors or because of a RATA Status error listed on this report.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- NOX RATA Status Evaluation

Check Code: HOURCV-30

Check Name: Initialize SO2 Calculated Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for SO2.

Specifications:

Current DHV Parameter = "SO2"
Current DHV Record Valid = SO2 Derived Hourly Status
SO2 Calculated Adjusted Value = null
Calculated Moisture for SO2 = null
Current DHV Record = Current SO2 Derived Hourly Record

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ SO2 Calculation Verification

Check Code: HOURCV-31

Check Name: Initialize NOX Calculated Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for NOX.

Specifications:

Current DHV Parameter = "NOX"
Current DHV Record Valid = NOX Derived Hourly Status
NOX Calculated Adjusted Value = null
Calculated Moisture for NOX = null
Current DHV Record = Current NOx Rate Derived Hourly Record

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Calculation Verification

Check Code: HOURCV-32

Check Name: Initialize NOXR Calculated Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for NOXR.

Specifications:

Current DHV Parameter = "NOXR"
Current DHV Record Valid = NOXR Derived Hourly Status
NOXR Calculated Adjusted Value = null
Calculated Diluent for NOXR = null
Calculated Moisture for NOXR = null
Current DHV HBHA Value = Current NOXR HBHA Value
Current DHV Record = Current NOx Rate Derived Hourly Record
Current Appendix E Status = null
RATA Status Required = false
RATA Status BAF = null
Current Hourly Record for RATA Status = ***Current NOx Rate Derived Hourly Record***

Set ***QaStatusComponentId*** = null
Set ***QaStatusComponentIdentifier*** = null
Set ***QaStatusComponentTypeCode*** = null
Set ***QaStatusSystemDesignationCode*** = ***CurrentDHVRecord***.SystemDesignationCode
Set ***QaStatusSystemId*** = ***CurrentDHVRecord***.SystemId
Set ***QaStatusSystemIdentifier*** = ***CurrentDHVRecord***.SystemIdentifier
Set ***QaStatusSystemTypeCode*** = ***CurrentDHVRecord***.SystemTypeCode

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification

Check Code: HOURCV-33

Check Name: Initialize CO2 Calculated Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for CO2.

Specifications:

Current DHV Parameter = "CO2"
Current DHV Record Valid = CO2 Derived Hourly Status
CO2 Calculated Adjusted Value = null
Calculated Diluent for CO2 = null
Calculated Moisture for CO2 = null
Current DHV Record = Current CO2 Mass Derived Hourly Record

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Mass Rate Calculation Verification

Check Code: HOURCV-34

Check Name: Initialize CO2C Calculated Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for CO2C.

Specifications:

Current DHV Parameter = "CO2C"
Current DHV Record Valid = CO2C Derived Hourly Status
CO2C DHV Calculated Adjusted Value = null
Calculated Diluent for CO2C = null
Calculated Moisture for CO2C = null
Current DHV HBHA Value = Current CO2C DHV HBHA Value
Current DHV Record = Current CO2 Conc Derived Hourly Record

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Calculation Verification

Check Code: HOURCV-35

Check Name: Initialize H2O Calculated Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for CO2C.

Specifications:

Current DHV Parameter = "H2O"
Current DHV Record Valid = H2O Derived Hourly Status
H2O DHV Calculated Adjusted Value = null
Current DHV HBHA Value = Current H2O DHV HBHA Value
Current DHV Record = Current H2O Derived Hourly Record

Set ***QaStatusComponentId*** = null
Set ***QaStatusComponentIdentifier*** = null
Set ***QaStatusComponentTypeCode*** = null
Set ***QaStatusSystemDesignationCode*** = ***CurrentDHVRecord***.SystemDesignationCode
Set ***QaStatusSystemId*** = ***CurrentDHVRecord***.SystemId
Set ***QaStatusSystemIdentifier*** = ***CurrentDHVRecord***.SystemIdentifier
Set ***QaStatusSystemTypeCode*** = ***CurrentDHVRecord***.SystemTypeCode

if (***Current DHV Record***.MODCCode in set {01, 02, 03, 21, 53} AND ***Current DHV Record***.SystemTypeCode == "H2O")
    RATA Status Required = true

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ H2O Calculation Verification

Check Code: HOURCV-36

Check Name: Initialize HI Calculated Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for HI.

Specifications:

Current DHV Parameter = "HI"
Current DHV Record Valid = HI Derived Hourly Status
HI Calculated Adjusted Value = null
Calculated Diluent for HI = null
Calculated Moisture for HI = null
Current DHV Record = Current Heat Input Derived Hourly Record
RATA Status Required = false
Current Hourly Record for RATA Status = Current Heat Input Derived Hourly Record

Set ***QaStatusComponentId*** = null
Set ***QaStatusComponentIdentifier*** = null
Set ***QaStatusComponentTypeCode*** = null
Set ***QaStatusSystemDesignationCode*** = ***CurrentDHVRecord***.SystemDesignationCode
Set ***QaStatusSystemIdentifier*** = ***CurrentDHVRecord***.SystemIdentifier
Set ***QaStatusSystemTypeCode*** = ***CurrentDHVRecord***.SystemTypeCode

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Heat Input Calculation Verification

Check Code: HOURCV-37

Check Name: Check Unadjusted Value

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that the UnadjustedValue in the DHV record for NOXR is valid.

Specifications:

Derived Hourly Unadjusted Value Status = false

If (***Current NOx Rate Method Code*** in set {CEM, PEM})

If (***Current DHV Record.***ModcCode in set {01, 02, 03, 04, 14, 21, 22, 53, 54})

If (***Current DHV Record***.UnadjustedHourlyValue is not null)

If (***Current DHV Record***.UnadjustedHourlyValue < 0.0 AND
Current DHV Record.ModcCode <> "21")
return result A

Else if (***Current DHV Record***.UnadjustedHourlyValue > 0 AND
Current DHV Record.ModcCode == 21)
return result B

Else if (***Current DHV Record***.UnadjustedHourlyValue is not rounded to three decimal places)
return result F

Else
Derived Hourly Unadjusted Value Status = true

if (***Current DHV Max Min Value*** is not null AND (***NOx Conc MODC*** is null OR is NOT in set {19, 20}))
if (***Current DHV Record***.UnadjustedHourlyValue > ***Current DHV Max Min Value***)
return result C

Else If (***Current DHV Record.***ModcCode not in set {04, 53, 54})
return result A
Else
Derived Hourly Unadjusted Value Status = true

Else if (***Derived Hourly Modc Status*** == true)

If (***Current DHV Record***.UnadjustedHourlyValue is not null)
return result D
Else
Derived Hourly Unadjusted Value Status = true
Else If (***Current DHV Record***.UnadjustedHourlyValue is not null)
return result E
Else
Derived Hourly Unadjusted Value Status = true

Results:

Result	Response	Severity
A	The UnadjustedHourlyValue reported in the DHV record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
B	You reported an MODCCode of 21 in the DHV record for [param], but the UnadjustedHourlyValue is greater than 0.	Critical Error Level 1
C	Warning: The UnadjustedHourlyValue reported in the DHV record for [param] is in excess of the maximum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update Span and/or Default values. You should investigate the cause of these exceedances and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Informational Message
D	You reported an MODCCode of [modcCode] in the DHV record for [param], so you should not have reported a value for the UnadjustedHourlyValue.	Critical Error Level 1
E	You reported an UnadjustedHourlyValue in the DHV record for [param]. A value in this field should not be reported for the [param] [method] method.	Critical Error Level 1
F	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification

Check Code: HOURCV-38

Check Name: Determine Maximum or Minimum Value for Parameter in DHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check determines the maximum or minimum value for the parameter from the span or default table based on MODC.

Specifications:

Current DHV Max Min Value = null

If (***Current DHV Record Valid*** == true AND
((***Current DHV Parameter*** == "H2O" AND ***H2O Method Code*** == "MWD") OR
(***Current DHV Parameter*** == "NOXR" AND ***Current NOx Rate Method Code*** in set {CEM,PEM}) OR
Current DHV Parameter == "CO2C" OR
Current DHV Parameter =="HI")

If (***Current DHV Parameter*** == "H2O")
If (***H2O Fuel Specific Missing Data*** == true)
Current DHV Fuel Specific Hour = true

If (***H2O Missing Data Approach*** == "MAX")
Current DHV Default Parameter = "H2OX"
Else If (***H2O Missing Data Approach*** == "MIN")
Current DHV Default Parameter = "H2ON"
Else If (***Current DHV Record.***ModcCode == 12)
return result A

else if (***Current DHV Parameter*** == "NOXR")
Current DHV Default Parameter = "NORX"

If (***Current DHV Record.***ModcCode in set {23, 24})
If (***NOx Rate Bypass Code*** == "BYMAXFS")
Current DHV Fuel Specific Hour = true
else
Current DHV Fuel Specific Hour = false
else if (***NOx Rate Fuel Specific Missing Data*** == true)
Current DHV Fuel Specific Hour = true
else
Current DHV Fuel Specific Hour = false

else if (***Current DHV Parameter*** == "CO2C")
Current DHV Default Parameter = "CO2X"

If (***CO2 Fuel Specific Missing Data*** == true)
Current DHV Fuel Specific Hour = true

If (***Current DHV Parameter*** == "HI")

Locate all active UnitCapacity records linked to the location where MaxHourlyHeatInputCapacity > 0.

If any are found,
Set ***Current DHV Max Min Value*** to the sum of MaxHourlyHeatInputCapacity in all records found.

else if (***Current DHV Default Parameter*** is not null)

If (***Current DHV Record.***ModcCode in set {12, 23, 25} AND ***Current DHV Fuel Specific Hour*** = true)

```
If (Current Hourly Op Record.FuelCode is not null)

    Current DHV Missing Data Fuel = Current Hourly Op Record.FuelCode

    Count active MonitoringDefaultData record for location where
        ParameterCode = Current DHV Default Parameter
        FuelCode = Current Hourly Op Record.FuelCode
        DefaultPurposeCode = "MD"     // Missing Data
        OperatingCode in set {A,U}      // Not Controlled

    if (count > 1)
        return result B
    else if (count == 0)
        return result C
    else
        Default Record = the single matched record

        if (Default Record.DefaultValue > 0)
            Current DHV Max Min Value = Default Record.DefaultValue
        else
            return result D

else if (Current DHV Record.ModcCode in set {13, 24}

    If (Current DHV Fuel Specific Hour == true)

        If Current Hourly Op Record.FuelCode is not null

            Current DHV Missing Data Fuel = Current Hourly Op Record.FuelCode

            Count active MonitoringDefaultData record for location where
                ParameterCode = Current DHV Default Parameter
                FuelCode = Current Hourly Op Record.FuelCode
                DefaultPurposeCode = "MD"     // Missing Data
                OperatingCode in == "C"      // Controlled

            if (count > 1)
                return result B
            else if (count == 0)
                return result C
            else
                Default Record = the single matched record

                if (Default Record.DefaultValue > 0)
                    Current DHV Max Min Value = Default Record.DefaultValue
                else
                    return result D
    else
        Current DHV Missing Data Fuel = "NFS"

        Count active MonitoringDefaultData record for location where
            ParameterCode = Current DHV Default Parameter
            FuelCode = "NFS"
            DefaultPurposeCode = "MD"     // Missing Data
            OperatingCode in == "C"      // Controlled
```

```
                    if (count > 1)
                        return result B
                    else if (count == 0)
                        return result C
                    else
                        Default Record = the single matched record

                        if (Default Record.DefaultValue > 0)
                            Current DHV Max Min Value = Default Record.DefaultValue
                        else
                            return result D

else if (Current DHV Record.ModcCode <> "15"

        Current DHV Missing Data Fuel = "NFS"

        Count active MonitoringDefaultData record for location where
            ParameterCode = Current DHV Default Parameter
            FuelCode = "NFS"
            DefaultPurposeCode = "MD"     // Missing Data
            OperatingCode in set {A,U}      // Not Controlled

        if (count > 1)
            return result B

        else if (count == 0 AND Current DHV Parameter == "CO2C")

            Monitor Span Record Count = Find active MonitoringSpanData records for location where
                    MonitoringSpanData .ComponentTypeCode = "CO2" AND
                    MonitoringSpanData.SpanScaleCode = "H"

            if (Monitor Span Record Count > 1)
                return result E
            else if (Monitor Span Record Count = 0)
                return result F
            else
                Current Monitor Span Record = the single matched record

                If (Current Monitor Span Record.DefaultHighRange is null AND Current DHV
                Record.ModcCode not in set {13, 24})

                    if (Current Monitor Span Record.MPCValue > 0)
                        Current DHV Max Min Value = Current Monitor Span Record.MPCValue
                    else
                        return result G

        else if (count == 0 ANDCurrent DHV Parameter == "NOXR")

            Count active MonitoringDefaultData record for location where
                ParameterCode = "MNNX"
                FuelCode = "NFS"
                DefaultPurposeCode = "MD"     // Missing Data
                OperatingCode in set {A,U}      // Not Controlled

             if (count > 1)
                Current DHV Default Parameter = "MNNX"
                return result B
```

```
                else if (count = 0)
                        return result C

                else
                        Current DHV Default Parameter = "MNNX"
                        Default Record = the single matched record

                        if (Default Record.DefaultValue >= 0)
                                Current DHV Max Min Value = Default Record.DefaultValue
                        else
                                return result D

        else if (count == 0)
                        return result C

        else
                Default Record = the single matched record

                if (Default Record.DefaultValue > 0)
                        Current DHV Max Min Value = Default Record.DefaultValue
                else
                        return result D
```

Results:

Result	Response	Severity
A	The missing data default parameter for H2O could not be determined, because you used both Standard and Inverse Part 75 missing data approaches during the hour.	Critical Error Level 2
B	You reported more than one applicable [param] Default record with a FuelCode of [FuelCode] in your monitoring plan for the hour.	Critical Error Level 1
C	You did not report an applicable [param] Default record with a FuelCode of [FuelCode].	Critical Error Level 1
D	The values reported in the applicable [param] Default record with a FuelCode of [FuelCode] are invalid.	Critical Error Level 1
E	You reported more than one active span record for [key] in your monitoring plan for the hour.	Critical Error Level 1
F	You did not report a missing data maximum default for CO2 in a span or default record in your monitoring plan.	Critical Error Level 1
G	The values reported in the applicable span record for [key] are invalid.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ H2O Calculation Verification
3	Process/Category:	Emissions Data Evaluation Report ------ Heat Input Calculation Verification
4	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification

Check Code: HOURCV-39

Check Name: Check Adjusted Hourly Value in DHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that AdjustedHourlyValue is valid and does not conflict with the reported MODC codes.

Validation Tables:

Parameter UOM (Complex Lookup Table)

Specifications:

Derived Hourly Adjusted Value Status = false
Current DHV Calculated Adjusted Value = null

if (***Current DHV Record Valid*** == true)

Locate ***Parameter Units of Measure*** lookup table record where ParameterCode = ***Current DHV Parameter***.

If (***Hourly Fuel Flow Count For Gas*** is greater than 0 AND ***Current DHV Parameter*** == "SO2")
Set *Current DHV Precision* to 4.

else
Set *Current DHV Precision* to the ***Parameter Units of Measure***.Decimals_Hrly.

if (***Current DHV Record***.ModcCode is not null)

case (***Current DHV Record.***ModcCode)

= 21: ***Current DHV Calculated Adjusted Value*** = 0
if (***Current DHV Record***.AdjustedHourlyValue == 0)
Derived Hourly Adjusted Value Status = true
else
return result A

= 12 OR 23 OR 25:
If (***Current DHV Max Min Value*** is not null)
Current DHV Calculated Adjusted Value = ***Current DHV Max Min Value***
if (***Current DHV Record***.AdjustedHourlyValue == ***Current DHV Max Min Value***)
Derived Hourly Adjusted Value Status = true
else
return result B

= 13 OR 24:
If (***Current DHV Max Min Value*** is not null)
Current DHV Calculated Adjusted Value = ***Current DHV Max Min Value***
if (***Current DHV Record***.AdjustedHourlyValue == ***Current DHV Max Min Value***)
Derived Hourly Adjusted Value Status = true
else
return result C

= 06:
If (***Current DHV Parameter*** in set {CO2C, H2O} AND (***Current DHV Record***.AdjustedHourlyValue is null or
Current DHV Record.AdjustedHourlyValue < 0 or ***Current DHV Record***.AdjustedHourlyValue > 100))
return result L

else if (***Current DHV HBHA Value*** is not null)

Current DHV Calculated Adjusted Value = ***Current DHV HBHA Value***

If (***Current DHV Record***.AdjustedHourlyValue >= 0)
if (***Current DHV Record***.AdjustedHourlyValue == ***Current DHV Calculated Adjusted Value***)
Derived Hourly Adjusted Value Status = true
else
return result D
else
return result E

else
If (***Current DHV Record***.AdjustedHourlyValue >= 0)
If (***Current DHV Record***.AdjustedHourlyValue is not rounded to *Current DHV Precision*)
return result M
else
Current DHV Calculated Adjusted Value = ***Current DHV Record***.AdjustedHourlyValue
Derived Hourly Adjusted Value Status = true

if (***Current DHV Parameter*** in set {CO2C, H2O, NOXR} AND ***Current DHV Max Min Value*** is not null)
If (***Current DHV Parameter*** == "H2O" AND ***H2O Missing Data Approach*** == "MIN")
if (***Current DHV Record***.AdjustedHourlyValue < ***Current DHV Max Min Value***)
return result H
else
if (***Current DHV Record***.AdjustedHourlyValue > ***Current DHV Max Min Value***)
If (***Current DHV Parameter*** == "NOXR" and ***Current DHV Record***.AdjustedHourlyValue > ***Current DHV Max Min Value*** * 2)
return result O
Otherwise,
return result G
Else
return result E

= 08 OR 09:

If (***Current DHV Parameter*** in set {CO2C, H2O} AND (***Current DHV Record***.AdjustedHourlyValue is null or ***Current DHV Record***.AdjustedHourlyValue < 0 or ***Current DHV Record***.AdjustedHourlyValue > 100))
return result L

else if (***Current DHV Record***.AdjustedHourlyValue >= 0)

If (***Current DHV HBHA Value*** is not null AND ***Current DHV Parameter*** == "H2O" AND ***H2O Missing Data Approach*** == "MIN" AND ***Current DHV HBHA Value*** < ***Current DHV Record***.AdjustedHourlyValue)
Current DHV Calculated Adjusted Value = ***Current DHV HBHA Value***
return result N

else if (***Current DHV HBHA Value*** is not null AND (***Current DHV Parameter*** <> "H2O" OR ***H2O Missing Data Approach*** == "MAX") AND ***Current DHV HBHA Value*** > ***Current DHV Record***.AdjustedHourlyValue AND (***Unit is Load Based*** == true or ***Current DHV Parameter*** <> "NOXR"))
Current DHV Calculated Adjusted Value = ***Current DHV HBHA Value***

return result F

elseif (***Current DHV Record***.AdjustedHourlyValue is not rounded to *Current DHV Precision*)
return result M
else
Current DHV Calculated Adjusted Value = ***Current DHV Record***.AdjustedHourlyValue
Derived Hourly Adjusted Value Status = true

if (***Current DHV Parameter*** in set {CO2C, H2O, NOXR} AND ***Current DHV Max Min Value*** is not null)
If (***Current DHV Parameter*** == "H2O" AND ***H2O Missing Data Approach*** == "MIN")
if (***Current DHV Record***.AdjustedHourlyValue < ***Current DHV Max Min Value***)
return result H
else
if (***Current DHV Record***.AdjustedHourlyValue > ***Current DHV Max Min Value***)
If (***Current DHV Parameter*** == "NOXR" and ***Current DHV Record***.AdjustedHourlyValue > ***Current DHV Max Min Value*** * 2)
return result O
Otherwise,
return result G

Else
return result E

= 04, 05, 07, 10, 11, 15, 53, 54, OR 55:

If (***Current DHV Parameter*** in set {CO2C, H2O} AND (***Current DHV Record***.AdjustedHourlyValue is null or ***Current DHV Record***.AdjustedHourlyValue < 0 or ***Current DHV Record***.AdjustedHourlyValue > 100))
return result L

else if (***Current DHV Record***.AdjustedHourlyValue >= 0)
If (***Current DHV Record***.AdjustedHourlyValue is not rounded to *Current DHV Precision*)
return result M
else
Current DHV Calculated Adjusted Value = ***Current DHV Record***.AdjustedHourlyValue
Derived Hourly Adjusted Value Status = true

if (***Current DHV Parameter*** in set {CO2C, H2O, NOXR} AND ***Current DHV Max Min Value*** is not null)
If (***Current DHV Parameter*** == "H2O" AND ***H2O Missing Data Approach*** == "MIN")
if (***Current DHV Record***.AdjustedHourlyValue < ***Current DHV Max Min Value***)
return result H
else
if (***Current DHV Record***.AdjustedHourlyValue > ***Current DHV Max Min Value***)
If (***Current DHV Parameter*** == "NOXR" and ***Current DHV Record***.AdjustedHourlyValue > ***Current DHV Max Min Value*** * 2)
return result O
Otherwise,
return result G
Else
return result E

= 26: If (***Current DHV Record***.AdjustedHourlyValue == 1)
Derived Hourly Adjusted Value Status = true
else
return result I

= All Other Codes except 40:

If (***Current DHV Parameter*** in set {CO2C, H2O} AND (***Current DHV Record***.AdjustedHourlyValue is null or ***Current DHV Record***.AdjustedHourlyValue < 0 or ***Current DHV Record***.AdjustedHourlyValue > 100))
return result L

else if (***Current DHV Record***.AdjustedHourlyValue >= 0)
If (***Current DHV Record***.AdjustedHourlyValue is not rounded to *Current DHV Precision*)
return result M

else
Derived Hourly Adjusted Value Status = true

If (***Current DHV Parameter*** in set {CO2C, H2O} AND ***Current DHV Max Min Value*** is not null)
If (***Current DHV Parameter*** == "H2O" AND ***H2O Missing Data Approach*** == "MIN")
if (***Current DHV Record***.AdjustedHourlyValue < ***Current DHV Max Min Value***)
return result H
else
if (***Current DHV Record***.AdjustedHourlyValue > ***Current DHV Max Min Value***)
return result G
Else
return result E
else

If (***Current DHV Record***.AdjustedHourlyValue >= 0)
If (***Current DHV Record***.AdjustedHourlyValue is not rounded to *Current DHV Precision*)
return result M

else

Derived Hourly Adjusted Value Status = true

If (***Current DHV Parameter*** == "HI")
if (***Heat Input Method Code*** not in set {AD, ADCALC, CALC})
Current DHV Calculated Adjusted Value = ***Current DHV Record***.AdjustedHourlyValue

If (***Current DHV Record***.AdjustedHourlyValue == 0.0)

If (***Heat Input Method Code*** == "CEM")

If ***Legacy Data Evaluation*** = true
If (***Current Hourly Op Record***.OpTime > 0.25)
return result J
else
If (***Current Hourly Op Record***.OpTime > 0)
return result K

else if (***Current DHV Max Min Value*** is not null and ***Current DHV Record***.AdjustedHourlyValue > ***Current DHV Max Min Value***)

```
                                    return result  G

                        else if (Current DHV Parameter == "NOXR")
                                if (Current NOx Rate Method Code <> "AE")
                                        Current DHV Calculated Adjusted Value = Current DHV Record.AdjustedHourlyValue

                        else if (Current DHV Parameter == "SO2")
                                if (SO2 App D Method Active for Hour == false)
                                        Current DHV Calculated Adjusted Value = Current DHV Record.AdjustedHourlyValue

                        else if (Current DHV Parameter == "CO2")
                                if (CO2 App D Method Active for Hour == false)
                                        Current DHV Calculated Adjusted Value = Current DHV Record.AdjustedHourlyValue
                        else
                                Current DHV Calculated Adjusted Value = Current DHV Record.AdjustedHourlyValue
else
        return result E
```

Results:

Result	Response	Severity
A	You reported an MODCCode of 21 in the DHV record for [param], but the AdjustedHourlyValue does not equal 0.	Critical Error Level 1
B	You reported an MODCCode of [modcCode] in the DHV record for [param], but the AdjustedHourlyValue does not equal the maximum potential value reported in the span or default record in your monitoring plan.	Critical Error Level 1
C	You reported an MODCCode of 13 or 24 in the DHV record for NOXR, but the AdjustedHourlyValue does not equal the maximum controlled emission rate reported in the NORX default record in your monitoring plan.	Critical Error Level 1
D	You reported an MODCCode of 06 in the DHV record for [param], but the AdjustedHourlyValue does not equal average of measured hour before and measured hour after.	Critical Error Level 1
E	The AdjustedHourlyValue reported in the DHV record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
F	You reported an MODCCode of [MODCCode] in the DHV record for [param], but you reported an AdjustedHourlyValue that is less than the average of the measured hour before and measured hour after.	Critical Error Level 1
G	Warning: The AdjustedHourlyValue reported in the DHV record for [param] is in excess of the maximum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update Span, Default, and/or Unit Capacity values. You should investigate the cause of these exceedances and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Informational Message
H	Warning: The AdjustedHourlyValue reported in the DHV record for [param] is lower than the minimum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these minimum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update Span Default values. You should investigate the cause of these low values and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Informational Message
I	You reported an MODCCode of 26 in the DHV record for [param], but the AdjustedHourlyValue does not equal 1.	Critical Error Level 1
J	You reported an AdjustedHourlyValue of 0 in the DHV record for HI, but you operated more than 0.25 hour.	Non-Critical Error
K	You reported an AdjustedHourlyValue of 0 in the DHV record for HI, but you had operating time during the hour. If you operated, you must report a heat input rate of at least 1 mmBtu/hr.	Critical Error Level 1
L	The AdjustedHourlyValue reported in the DHV record for [param] is invalid. The value must be between 0 and 100.	Critical Error Level 1
M	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
N	You reported an MODCCode of [MODCCode] in the DHV record for [param], but you reported an AdjustedHourlyValue that is greater than the average of the measured hour before and measured hour after.	Critical Error Level 1
O	The AdjustedHourlyValue reported in the DHV record for NOXR is in excess of 200% of the maximum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update Span and/or Default values. You should investigate the cause of these exceedances and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Critical Error Level 2

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ CO2 Mass Rate Calculation Verification
3	Process/Category:	Emissions Data Evaluation Report ------ H2O Calculation Verification
4	Process/Category:	Emissions Data Evaluation Report ------ Heat Input Calculation Verification
5	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification
6	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Calculation Verification
7	Process/Category:	Emissions Data Evaluation Report ------ SO2 Calculation Verification

Check Code: HOURCV-40

Check Name: Determine Moisture for SO2 Mass Calculation Verification

Related Former Checks:

Applicability: CEM Check

Description: Verifies that all elements are present to support the equation code indicated by the current SO2 Mass Rate record

Specifications:

If (***SO2 Method Code*** begins with "CEM")
if (***SO2 Mass Equation Code*** == "F-2")
If (***H2O Method Code*** == "MWD" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
Calculated Moisture for SO2 = H2O DHV Calculated Adjusted Value
else if (***H2O Method Code*** in set {MMS, MTB} AND ***H2O Monitor Hourly Checks Needed*** == true AND ***H2O MHV Calculated Adjusted Value*** is not null)
Calculated Moisture for SO2 = H2O MHV Calculated Adjusted Value
else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == true AND ***H2O DHV Calculated Adjusted Value*** is not null)
Calculated Moisture for SO2 = H2O DHV Calculated Adjusted Value
else if (***H2O Method Code*** == "MDF" AND ***H2O Derived Hourly Checks Needed*** == false AND ***H2O Default Value*** is not null)
Calculated Moisture for SO2 = H2O Default Value

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ SO2 Calculation Verification

Check Code: HOURCV-41

Check Name: Calculate Adjusted NOx Rate in DHV Record

Related Former Checks:

Applicability: General Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

if (***Derived Hourly Adjusted Value Status*** == true)

if (***Rpt Period NOx Rate Reported Accumulator Array*** for this location is not null)
if (***Rpt Period NOx Rate Reported Accumulator Array*** for this location >= 0)
Rpt Period NOx Rate Reported Accumulator Array for this location = ***Rpt Period NOx Rate Reported Accumulator Array*** for this location + ***Current DHV Record***.AdjustedHourlyValue
else
Rpt Period NOx Rate Reported Accumulator Array for this location = ***Current DHV Record***.AdjustedHourlyValue
else
Rpt Period NOx Rate Reported Accumulator Array for this location = -1

if (***RATA Status Required*** == true AND ***Current NOX System BAF*** is not null)
NOXR Calculated Adjusted Value = ***NOXR Calculated Unadjusted Value*** * ***Current NOX System BAF***, and round the result to three decimal places

else if (***Current NOx Rate Method Code*** == "AE" AND ***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil*** > 0 AND ***App E Constant Fuel Mix*** == false)

If (***NOXR App E Accumulator*** >= 0 AND ***HI Calculated Adjusted Value*** is not null AND ***Current Hourly Op Record***.OperatingTime is between 0 and 1 (inclusive))

If (***NOXR App E Accumulator*** == 0)
NOXR Calculated Adjusted Value = 0
else
NOXR Calculated Adjusted Value = ***NOXR App E Accumulator*** / ***Total Heat Input from Fuel Flow***, and round the result to three decimal places
else
Apportionment Calc NOXR Array at this Location = -1
Rpt Period NOx Rate Calculated Accumulator Array for this location = -1
return result A

If (***NOXR Calculated Adjusted Value*** is not null)
Apportionment Calc NOXR Array at this Location = ***NOXR Calculated Adjusted Value***

if (***MP Stack Config for Hourly Checks*** == "MS" AND ***Expected Summary Value NOx Rate Array*** for the location == true)
if (***Config NOxRateTimesHeatInput Accumulator*** >= 0 AND ***HI Calculated Adjusted Value*** is not null)
Config NOxRateTimesHeatInput Accumulator = ***Config NOxRateTimesHeatInput Accumulator*** + (***HI Calculated Adjusted Value*** * ***NOXR Calculated Adjusted Value***)
else
Config NOxRateTimesHeatInput Accumulator = -1

if (***Config NOxRateTimesOpTime Accumulator*** >= 0 AND ***Current Hourly Op Record***.OperatingTime is between 0 and 1 (inclusive))
Config NOxRateTimesOpTime Accumulator = ***Config NOxRateTimesOpTime Accumulator*** + (***Current Hourly Op Record***.OperatingTime * ***NOXR Calculated Adjusted Value***)
Config OpTime Accumulator = ***Config OpTime Accumulator*** + ***Current Hourly Op Record***.OperatingTime

else
Config NOxRateTimesOpTime Accumulator = -1

if (***Rpt Period NOx Rate Calculated Accumulator Array*** for this location is not null)
if (***Rpt Period NOx Rate Calculated Accumulator Array*** for this location >= 0)
Rpt Period NOx Rate Calculated Accumulator Array for this location = ***Rpt Period NOx Rate Calculated Accumulator Array*** for this location + ***NOXR Calculated Adjusted Value***
else
Rpt Period NOx Rate Calculated Accumulator Array for this location = ***NOXR Calculated Adjusted Value***

Rpt Period NOx Rate Hours Accumulator Array for this location = ***Rpt Period NOx Rate Hours Accumulator Array*** for this location + 1

if (***Derived Hourly Adjusted Value Status*** == true AND ***Derived Hourly Unadjusted Calculation Status*** == true)

Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "NOXR" AND
UOM = "LBMMBTU"

if ABS(***NOXR Calculated Adjusted Value*** - ***Current DHV Record***.AdjustedHourlyValue) > *Tolerance*
return result B

else if ***Current NOx Rate Method Code*** <> "AE" OR ***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil*** > 0)
> 0)
Apportionment Calc NOXR Array at this Location = -1
Rpt Period NOx Rate Calculated Accumulator Array for this location = -1

if (***MP Stack Config for Hourly Checks*** == "MS")
Config NOxRateTimesHeatInput Accumulator = -1

if (***RATA Status Required*** == true AND ***Current NOX System BAF*** is null AND ***NOXR Calculated Unadjusted Value*** is not null
)
return result A

Results:

Result	Response	Severity
A	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated due to errors listed above.	Informational Message
B	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification
	Conditions:	RATA Status Required Equals false
2	Process/Category:	Emissions Data Evaluation Report --------- NOX RATA Status Evaluation

Check Code: HOURCV-42

Check Name: Check HI System in DHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that a valid Monitoring System is indicated in the DHV record.

Specifications:

If Current DHV Parameter == "HI" AND ***Heat Input Method Code*** in set {CEM, AMS})

If (***Current DHV Record***.MonitoringSystemID is null

If (***Heat Input Method Code*** == "CEM")

If (***CO2 Conc Checks Needed for Heat Input*** == true AND ***Current CO2 Conc Monitor Hourly Record*** is not null AND ***Current CO2 Conc Monitor Hourly Record.***ModcCode in set {01, 02, 03, 04, 17, 20, 21}) OR
(***O2 Wet Checks Needed for Heat Input*** == true AND ***Current O2 Wet Monitor Hourly Record*** is not null AND ***Current O2 Wet Monitor Hourly Record.***ModcCode in set {01, 02, 03, 04, 17, 20}) OR
(***O2 Dry Checks Needed for Heat Input*** == true AND ***Current O2 Dry Monitor Hourly Record*** is not null AND ***Current O2 Dry Monitor Hourly Record.***ModcCode in set {01, 02, 03, 04, 17, 20})
return result A
Else
return result E

Else

HI DHV Mon Sys Record = find active MonitoringSystemData record for location where MonitoringSystemData.MonitoringSystemID = ***Current DHV Record***.MonitoringSystemID

If (*HI DHV Mon Sys Record* is null)
return result C

Else If (*HI DHV Mon Sys Record*.SystemTypeCode not in set {CO2, O2})
return result D

Else If (***Heat Input Method Code*** != "CEM") OR
(***CO2 Conc Checks Needed for Heat Input*** == true AND ***Current CO2 Conc Monitor Hourly Record*** is not null AND ***Current CO2 Conc Monitor Hourly Record.***ModcCode in set {01, 02, 03, 04, 17, 20, 21}) OR
(***O2 Wet Checks Needed for Heat Input*** == true AND ***Current O2 Wet Monitor Hourly Record*** is not null AND ***Current O2 Wet Monitor Hourly Record.***ModcCode in set {01, 02, 03, 04, 17, 20}) OR
(***O2 Dry Checks Needed for Heat Input*** == true AND ***Current O2 Dry Monitor Hourly Record*** is not null AND ***Current O2 Dry Monitor Hourly Record.***ModcCode in set {01, 02, 03, 04, 17, 20})

If (***CO2 RATA Required*** == true)
RATA Status Required = true

Results:

Result	Response	Severity
A	You did not report MonitoringSystemID in the DHV record for HI. While this was acceptable for legacy data, this field is required when you report measured data.	Critical Error Level 1
B	You reported a MonitoringSystemID in the DHV record for [param]. This field should be blank when missing data substitution is used.	Non-Critical Error
C	You reported MonitoringSystemID [ID] in the DHV record for [param], but according to your monitoring plan this system was not active during the hour.	Critical Error Level 1
D	You reported MonitoringSystemID [ID] in the DHV record for HI, but this system is not a CO2 or O2 monitoring system.	Critical Error Level 1
E	You did not report MonitoringSystemID in the DHV record for HI. MonitoringSystemID is required when you report missing data.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Heat Input Calculation Verification

Check Code: HOURCV-43

Check Name: Determine DHV Measure Code

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Set ***Current Measure Code*** to null.

If (***Current DHV Parameter*** in set {CO2C, H2O})

If (***Current DHV Record.***ModcCode in set {01, 02, 03, 04, 21, 53, 54})
Set ***Monitor Measure Code Array*** for the ***Current DHV Parameter*** to "MEASURE"

if (***Current DHV Parameter*** == "CO2C" AND ***CO2 Conc CEM Equation Code*** == "F-14B" AND ***Monitor Measure Code Array*** for "H2O" == "SUB")
Set ***Monitor Measure Code Array*** for "CO2C" to "MEASSUB".

else if (***Current DHV Record.***ModcCode in set {06, 07, 08, 09, 10, 12, 55})
Set ***Monitor Measure Code Array*** for the ***Current DHV Parameter*** to "SUB"

if (***Current DHV Parameter*** == "CO2C" AND ***CO2 Conc CEM Equation Code*** == "F-14B" AND ***Monitor Measure Code Array*** for "H2O" == "MEASURE")
Set ***Monitor Measure Code Array*** for "CO2C" to "MEASSUB".

else if (***Current DHV Parameter*** == "NOXR")

If (***Current NOx Rate Method Code*** in set {CEM, PEM}

If (***Current DHV Record.***ModcCode in set {01, 02, 03, 04, 05, 14, 21, 22, 53, 54})
Set ***Current Measure Code***" to "MEASURE".

if (***NOx Rate Equation Code*** in set {19-3, 19-3D, 19-4, 19-5, 19-8, 19-9} AND ***Monitor Measure Code Array*** for "H2O" == "SUB")
set ***Current Measure Code*** to "MEASSUB".

else if (***Current DHV Record.***ModcCode in set {06, 07, 08, 09, 10, 11, 12, 13, 15, 23, 24, 25, 55})
Set ***Current Measure Code*** to "SUB"

if (***NOx Rate Equation Code*** in set {19-3, 19-3D, 19-4, 19-5, 19-8, 19-9} AND ***Monitor Measure Code Array*** for "H2O" == "MEASURE")
set ***Current Measure Code*** to "MEASSUB".

else if (***Current NOx Rate Method Code*** == "AE")
Set ***Current Measure Code*** to the ***Monitor Measure Code Array*** for "NOXR".

Set ***NOXR Measure Code*** to the ***Current Measure Code***.

else if (***Current DHV Parameter*** == "HI")

If (***Heat Input Method Code*** == "CEM")

if (***Heat Input Equation Code*** in set {F-15, F-16})

If (***Monitor Measure Code Array*** for "CO2C" and "FLOW" are both equal to "MEASURE")
set ***Current Measure Code*** to "MEASURE".

else if (***Monitor Measure Code Array*** for "CO2C" and "FLOW" are both equal to "SUB")
set ***Current Measure Code*** to "SUB".

else if (***Monitor Measure Code Array*** for "CO2C" and "FLOW" are both not null)
set ***Current Measure Code*** to "MEASSUB".

else if (***Heat Input Equation Code*** in set {F-18})

If (***Monitor Measure Code Array*** for "O2D" and "FLOW" are both equal to "MEASURE")
set ***Current Measure Code*** to "MEASURE".

else if (***Monitor Measure Code Array*** for "O2D" and "FLOW" are both equal to "SUB")
set ***Current Measure Code*** to "SUB".

else if (***Monitor Measure Code Array*** for "O2D" and "FLOW" are both not null)
set ***Current Measure Code*** to "MEASSUB".

else if (***Heat Input Equation Code*** in set {F-17})

If (***Monitor Measure Code Array*** for "O2W" and "FLOW" are both equal to "MEASURE")
set ***Current Measure Code*** to "MEASURE".

else if (***Monitor Measure Code Array*** for "O2W" and "FLOW" are both equal to "SUB")
set ***Current Measure Code*** to "SUB".

else if (***Monitor Measure Code Array*** for "O2W" and "FLOW" are both not null)
set ***Current Measure Code*** to "MEASSUB".

if (***Heat Input Equation Code*** in set {F-16, F-17, F-18} AND ***Monitor Measure Code Array*** for "H2O" is not null)

If ((***Current Measure Code*** == "MEASURE" AND ***Monitor Measure Code Array*** for "H2O" == "SUB") OR
(***Current Measure Code*** == "SUB" AND ***Monitor Measure Code Array*** for "H2O" == "MEASURE"))
set ***Current Measure Code*** to "MEASSUB".

else if (***Heat Input App D Method Active for Hour*** == true)

If (***Monitor Measure Code Array*** for "FF" in set {OTHER, MEASSUB})
set ***Current Measure Code*** to ***Monitor Measure Code Array*** for "FF".

else if (***Monitor Measure Code Array*** for "FF" and "GCV" are both equal to "MEASURE")
set ***Current Measure Code*** to "MEASURE".

else if (***Monitor Measure Code Array*** for "FF" and "GCV" are both equal to "SUB")
set ***Current Measure Code*** to "SUB".

else if (***Monitor Measure Code Array*** for "FF" and "GCV" are both not null)
set ***Current Measure Code*** to "MEASSUB".

if (***Monitor Measure Code Array*** for "DENSITY" is not null)

If ((***Current Measure Code*** == "MEASURE" AND ***Monitor Measure Code Array*** for "DENSITY" == "SUB")
OR (***Current Measure Code*** == "SUB" AND ***Monitor Measure Code Array*** for "DENSITY" == "MEASURE"))
set ***Current Measure Code*** to "MEASSUB".

Set ***HI Measure Code*** to the ***Current Measure Code***.

else if (***Current DHV Parameter*** == "SO2")

If (***SO2 CEM Method Active for Hour*** == true)

If (***Monitor Measure Code Array*** for "SO2C" and "FLOW" are both equal to "MEASURE")
set ***Current Measure Code*** to "MEASURE".

else if (***Monitor Measure Code Array*** for "SO2C" and "FLOW" are both equal to "SUB")
set ***Current Measure Code*** to "SUB".

else if (***Monitor Measure Code Array*** for "SO2C" and "FLOW" are both not null)
set ***Current Measure Code*** to "MEASSUB".

if (***SO2 Equation Code*** == "F-2" AND ***Monitor Measure Code Array*** for "H2O" is not null)

If ((***Current Measure Code*** == "MEASURE" AND ***Monitor Measure Code Array*** for "H2O" == "SUB") OR
(***Current Measure Code*** == "SUB" AND ***Monitor Measure Code Array*** for "H2O" == "MEASURE"))
set ***Current Measure Code*** to "MEASSUB".

else if (***SO2 F23 Method Active for Hour*** == true)

set ***Current Measure Code*** to ***HI Measure Code.***

else if (***SO2 App D Method Active for Hour*** == true)

If (***Monitor Measure Code Array*** for "FF" in set {OTHER, MEASSUB} OR ***Monitor Measure Code Array*** for "SULFUR" is null)
set ***Current Measure Code*** to ***Monitor Measure Code Array*** for "FF".

else if (***Monitor Measure Code Array*** for "FF" and "SULFUR" are both equal to "MEASURE")
set ***Current Measure Code*** to "MEASURE".

else if (***Monitor Measure Code Array*** for "FF" and "SULFUR" are both equal to "SUB")
set ***Current Measure Code*** to "SUB".

else if (***Monitor Measure Code Array*** for "FF" is not null)
set ***Current Measure Code*** to "MEASSUB".

else if (***Current DHV Parameter*** == "CO2")

If (***CO2 Method Code*** == "CEM")

If (***Monitor Measure Code Array*** for "CO2C" and "FLOW" are both equal to "MEASURE")
set ***Current Measure Code*** to "MEASURE".

else if (***Monitor Measure Code Array*** for "CO2C" and "FLOW" are both equal to "SUB")
set ***Current Measure Code*** to "SUB".

else if (***Monitor Measure Code Array*** for "CO2C" and "FLOW" are both not null)
set ***Current Measure Code*** to "MEASSUB".

if (***CO2 Mass Equation Code*** == "F-2" AND ***Monitor Measure Code Array*** for "H2O" is not null)

If ((***Current Measure Code*** == "MEASURE" AND ***Monitor Measure Code Array*** for "H2O" == "SUB") OR
(***Current Measure Code*** == "SUB" AND ***Monitor Measure Code Array*** for "H2O" == "MEASURE"))
set ***Current Measure Code*** to "MEASSUB".

else if (***CO2 App D Method Active for Hour*** == true)

set ***Current Measure Code*** to ***HI Measure Code.***

else if (***Current DHV Parameter*** == "NOX")

if (***NOx Mass Equation Code*** == "F-24A")

If (***HI Measure Code and NOXR Measure Code*** are both equal to "MEASURE")
set ***Current Measure Code*** to "MEASURE".

else if (***HI Measure Code and NOXR Measure Code*** are both equal to "SUB")
set ***Current Measure Code*** to "SUB".

else if (***HI Measure Code and NOXR Measure Code*** are both not null)
set ***Current Measure Code*** to "MEASSUB".

else if (***NOx Mass Equation Code*** in set {F-26A, F-26B})

If (***Monitor Measure Code Array*** for "NOXC" and "FLOW" are both equal to "MEASURE")
set ***Current Measure Code*** to "MEASURE".

else if (***Monitor Measure Code Array*** for "NOXC" and "FLOW" are both equal to "SUB")
set ***Current Measure Code*** to "SUB".

else if (***Monitor Measure Code Array*** for "NOXC" and "FLOW" are both not null)
set ***Current Measure Code*** to "MEASSUB".

if (***NOx Mass Equation Code*** == "F-26B" AND ***Monitor Measure Code Array*** for "H2O" is not null)

If ((***Current Measure Code*** == "MEASURE" AND ***Monitor Measure Code Array*** for "H2O" == "SUB") OR
(***Current Measure Code*** == "SUB" AND ***Monitor Measure Code Array*** for "H2O" == "MEASURE"))
set ***Current Measure Code*** to "MEASSUB".

else if (***Current DHV Parameter*** in set {SO2M, NOXM, CO2M, HIT}
Set ***Current Measure Code*** to "LME".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ CO2 Mass Rate Calculation Verification
3	Process/Category:	Emissions Data Evaluation Report ------ CO2M Derived Hourly Evaluation (LME)
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Calculation Verification
5	Process/Category:	Emissions Data Evaluation Report ------ Heat Input Calculation Verification
6	Process/Category:	Emissions Data Evaluation Report ------ HIT Derived Hourly Evaluation (LME)
7	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Calculation Verification
8	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Calculation Verification
9	Process/Category:	Emissions Data Evaluation Report ------ NOXM Derived Hourly Evaluation (LME)
10	Process/Category:	Emissions Data Evaluation Report ------ SO2 Calculation Verification
11	Process/Category:	Emissions Data Evaluation Report ------ SO2M Derived Hourly Evaluation (LME)

Check Category:

Hourly Derived Data

Check Code: HOURDHV-1

Check Name: Initialize SO2 Derived Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for SO2.

Specifications:

Current DHV Parameter = "SO2"
SO2 Derived Hourly Status = true
Current DHV Record = Current SO2 Derived Hourly Record
Current DHV Method = SO2 Method Code
Current DHV System Type = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ SO2 Derived Hourly Evaluation

Check Code: HOURDHV-2

Check Name: Initialize NOX Derived Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for NOX.

Specifications:

Current DHV Parameter = "NOX"
NOX Derived Hourly Status = true
Current DHV Record = Current NOx Mass Derived Hourly Record
Current DHV Method = NOx Mass Monitor Method Code
Current DHV System Type = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Derived Hourly Evaluation

Check Code: HOURDHV-3

Check Name: Initialize NOXR Derived Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for NOXR.

Specifications:

Current DHV Parameter = "NOXR"
NOXR Derived Hourly Status = true
Current DHV System Type = null
Current DHV Record = Current NOx Rate Derived Hourly Record
Current DHV Method = Current NOx Rate Method Code
NOx Emission Rate MODC = ***Current NOx Rate Derived Hourly Record.***ModcCode

if (***Current DHV Method*** == "CEM"***)***
    Current DHV System Type = "NOX"
else if (***Current DHV Method*** == "PEM")
    Current DHV System Type = "NOXP"
if (***Current DHV Method*** == "AE")
    Current DHV System Type = "NOXE"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation

Check Code: HOURDHV-4

Check Name: Initialize CO2 Derived Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for CO2.

Specifications:

Current DHV Parameter = "CO2"
CO2 Derived Hourly Status = true
Current DHV Record = Current CO2 Mass Derived Hourly Record
Current DHV Method = CO2 Method Code
Current DHV System Type = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Mass Rate Derived Hourly Evaluation

Check Code: HOURDHV-5

Check Name: Initialize CO2C Derived Hourly Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for CO2C.

Specifications:

Current DHV Parameter = "CO2C"
CO2C Derived Hourly Status = true
Current DHV Record = Current CO2 Conc Derived Hourly Record
Current DHV System Type = 'CO2'
Current DHV Method = "CEM"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Derived Hourly Evaluation

Check Code: HOURDHV-6

Check Name: Initialize H2O Derived Hourly Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for H2O.

Specifications:

Current DHV Parameter = "H2O"
H2O Derived Hourly Status = true
Current DHV Record = Current H2O Derived Hourly Record
Current DHV System Type = "H2O"
Current DHV Method = H2O Method Code
RATA Status Required = false
Current Hourly Record for RATA Status = ***Current H2O Derived Hourly Record***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation

Check Code: HOURDHV-7

Check Name: Initialize HI Derived Hourly Data

Related Former Checks:

Applicability: General Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for HI.

Specifications:

Current DHV Parameter = "HI"
HI Derived Hourly Status = true
Current DHV System Type = null
Current DHV Record = Current Heat Input Derived Hourly Record
Current DHV Method = Heat Input Method Code

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Heat Input Derived Hourly Evaluation

Check Code: HOURDHV-8

Check Name: Initialize SO2R Derived Hourly Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for SO2R.

Specifications:

Current DHV Parameter = "SO2R"
SO2R Derived Hourly Status = true
Current DHV System Type = null
Current DHV Record = Current SO2R Derived Hourly Record
Current DHV Method = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ SO2R Derived Hourly Evaluation

Check Code: HOURDHV-9

Check Name: Initialize SO2M Derived Hourly Data

Related Former Checks:

Applicability: LME Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for SO2M.

Specifications:

Current DHV Parameter = "SO2M"
SO2M Derived Hourly Status = true
Current DHV System Type = null
Current DHV Method = "LME"
Current DHV Record = ***Current SO2 Derived Hourly Record***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ SO2M Derived Hourly Evaluation (LME)

Check Code: HOURDHV-10

Check Name: Initialize NOXM Derived Hourly Data

Related Former Checks:

Applicability: LME Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for NOXM.

Specifications:

Current DHV Parameter = "NOXM"
NOXM Derived Hourly Status = true
Current DHV System Type = null
Current DHV Method = "LME"
Current DHV Record = ***Current NOx Mass Derived Hourly Record***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOXM Derived Hourly Evaluation (LME)

Check Code: HOURDHV-11

Check Name: Initialize CO2M Derived Hourly Data

Related Former Checks:

Applicability: LME Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for CO2M.

Specifications:

Current DHV Parameter = "CO2M"
CO2M Derived Hourly Status = true
Current DHV System Type = null
Current DHV Method = "LME"
Current DHV Record = ***Current CO2 Mass Derived Hourly Record***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2M Derived Hourly Evaluation (LME)

Check Code: HOURDHV-12

Check Name: Initialize HIT Derived Hourly Data

Related Former Checks:

Applicability: LME Check

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for HIT.

Specifications:

Current DHV Parameter = "HIT"
HIT Derived Hourly Status = true
Current DHV System Type = null
Current DHV Method = ***Heat Input Method Code***
Current DHV Record = ***Current Heat Input Derived Hourly Record***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ HIT Derived Hourly Evaluation (LME)

Check Code: HOURDHV-13

Check Name: Check MODC in DHV Record

Related Former Checks:

Applicability: CEM Check

Description: Basic check to ensure that MODC reported in the DHV record is valid for the parameter. Also initializes variables for the category.

Specifications:

Derived Hourly Modc Status = false

case (***Current DHV Parameter***)

SO2: If (***Current DHV Method*** == "AMS")
If (***Current DHV Record.***ModcCode is not null and is not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 12, 13, 15, 16, 21, 23, 53, 54, 55})
return result A
Else
Derived Hourly Modc Status = true
else
If (***Current DHV Record.***ModcCode is not null)
return result B
Else
Derived Hourly Modc Status = true

NOX: If (***Current DHV Method*** == "AMS")
If (***Current DHV Record.***ModcCode is not null and is not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 15, 21, 23, 24, 53, 54, 55})
return result A
Else
Derived Hourly Modc Status = true
else
If (***Current DHV Record.***ModcCode is not null)
return result B
Else
Derived Hourly Modc Status = true

NOXR: If (***Current DHV Method*** == "AMS" AND ***Current DHV Record.***ModcCode is null)
Derived Hourly Modc Status = true

else if (***Current DHV Method*** == "AE")
If (***Current DHV Record.***ModcCode is not null)
return result C
Else
Derived Hourly Modc Status = true
else
if (***Current DHV Record.***ModcCode not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 53, 54, 55})
return result A
Else
Derived Hourly Modc Status = true

CO2C: If (***Current DHV Record.***ModcCode not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 12, 21, 53, 54, 55})
return result A
Else
Derived Modc Status = true

CO2: If (***Current DHV Method*** == "AMS")
If (***Current DHV Record.***ModcCode is not null and is not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 12, 53, 54, 55})
return result A
Else
Derived Hourly Modc Status = true
else
If (***Current DHV Record.***ModcCode is not null)
return result B
Else
Derived Hourly Modc Status = true

HI: If (***Current DHV Method*** == "AMS")
If (***Current DHV Record.***ModcCode is not null and is not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 12, 26, 53, 54, 55})
return result A
Else
Derived Hourly Modc Status = true
else
If (***Current DHV Record.***ModcCode is not null and ***Current DHV Record.***ModcCode <> "26")
return result B
Else
Derived Hourly Modc Status = true

H2O: ***H2O DHV MODC*** = ***Current DHV Record.***ModcCode

if (***Current DHV Method*** == "MWD")
If (***Current DHV Record.***ModcCode not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 12, 21, 53, 54, 55})
return result A
else
Derived Hourly Modc Status = true

else if (***Current DHV Method*** == "MDF")
If (***Current DHV Record.***ModcCode <> "40")
return result A
else
Derived Hourly Modc Status = true

SO2R: If (***SO2 F23 Method Active For Hour*** == true)
If (***Current DHV Record.***ModcCode <> "40")
return result A
else
Derived Hourly Modc Status = true

HIT: If (***Current DHV Record.***ModcCode == "45")
If (***LME HI Substitute Data Code*** == "MHHI")
Derived Hourly Modc Status = true
else
return result D
else if (***Current DHV Record.***ModcCode is not null)
return result A
Else
Derived Hourly Modc Status = true

SO2M, NOXM, CO2M:

If (***Current DHV Record.***ModcCode is not null)
return result B
Else
Derived Hourly Modc Status = true

Results:

Result	Response	Severity
A	The MODCCode reported in the DHV record for [param] is invalid.	Critical Error Level 1
B	You reported an MODCCode in the DHV record for [param]. This field should be blank.	Non-Critical Error
C	You reported an MODCCode in the DHV record for NOXR. This field should be blank when you use the Appendix E method to determine the NOx emission rate.	Critical Error Level 1
D	You reported an MODCCode of 45 in the DHV record for HIT, but you have not reported a SubstituteDataCode of MHHI in the active heat input method record in your monitoring plan.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2 Mass Rate Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ CO2M Derived Hourly Evaluation (LME)
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ Heat Input Derived Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ HIT Derived Hourly Evaluation (LME)
7	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Derived Hourly Evaluation
9	Process/Category:	Emissions Data Evaluation Report ------ NOXM Derived Hourly Evaluation (LME)
10	Process/Category:	Emissions Data Evaluation Report ------ SO2 Derived Hourly Evaluation
11	Process/Category:	Emissions Data Evaluation Report ------ SO2M Derived Hourly Evaluation (LME)
12	Process/Category:	Emissions Data Evaluation Report ------ SO2R Derived Hourly Evaluation

Check Code: HOURDHV-14

Check Name: Check Percent Monitor Availability in DHV Record

Related Former Checks:

Applicability: CEM Check

Description: Performs a series basic checks to ensure that the reported monitor percent available is between 0 and 100, inclusive, then checks to see that percent available is within permitted ranges for specific MODC codes

Specifications:

Derived Hourly Pma Status = false
Derived Hourly Missing Data Status = true

```
If (Derived Hourly Modc Status == true)
    If ( Current DHV Record.PercentAvailable is NULL)
        if (Current DHV Parameter not in set {H2O, CO2C, NOXR})
            Derived Hourly Pma Status = true

        else if (Current DHV Parameter == "NOXR" and Current DHV Method not in set {PEM, CEM})
            Derived Hourly Pma Status = true

        else if (Current DHV Parameter == "H2O" and Current DHV Record.ModcCode == "40")
            Derived Hourly Pma Status = true

        else
            if (Current DHV Record.ModcCode not in set {01, 02, 03, 04, 14, 21, 22, 53, 54} AND Legacy Data
            Evaluation == true)
                Derived Hourly Pma Status = true
                return result A
            else
                return result B
    else
        if (Current DHV Parameter == "NOXR" and Current DHV Method == "AE")
            return result C

        else if (Current DHV Parameter == "H2O" and Current DHV Record.ModcCode == "40")
            return result C

        else if (Current DHV Parameter not in set {H2O, CO2C, NOXR} AND Current DHV Method <> "AMS")
            return result C

        else if (Current DHV Record.PercentAvailable> 100.0 OR
                 Current DHV Record.PercentAvailable < 0.0)
            return result D

        Else
            case ( Current DHV Record.ModcCode )
            = 06:  If  Current DHVRecord.PercentAvailable >= 90.0
                       Derived Hourly Pma Status = true
                   Else
                       return result E

            = 08:  If  Current DHV Record.PercentAvailable >= 95.0
                       Derived Hourly Pma Status = true
                   Else
                       return result E
            = 09:  If  Current DHV Record.PercentAvailable >= 90.0 AND Current DHV Record.PercentAvailable < 95.0
                       Derived Hourly Pma Status = true
```

Else
return result E
= 10: If ***Current DHV Record***.PercentAvailable >=80.0 AND ***Current DHV Record***.PercentAvailable < 90.0
Derived Hourly Pma Status = true
Else If ***Current DHV Parameter*** == "NOXR" and ***Current DHV Record***.PercentAvailable >=90.0
Derived Hourly Pma Status = true
return result F
Else
return result E
= 11: If ***Current DHV Record***.PercentAvailable >=90.0
Derived Hourly Pma Status = true
Else
return result E

All other MODC Codes:
Derived Hourly Pma Status = true

Results:

Result	Response	Severity
A	You reported an MODCCode of [ModcCode] in the DHV record for [param], but you did not report a value for PercentAvailable. While this is not required for legacy EDR data, it is required in all [param] DHV records for ECMPS.	Informational Message
B	You did not report PercentAvailable in the DHV record for [param].	Critical Error Level 1
C	You reported PercentAvailable in the DHV record for [param]. This field should be blank.	Critical Error Level 1
D	The PercentAvailable reported in the DHV record for [param] is invalid. This value must be between 0 and 100.	Critical Error Level 1
E	You reported an MODCCode of [modcCode] in the DHV for [param], but the PercentAvailable is not appropriate for this MODC.	Critical Error Level 1
F	You reported an MODCCode of 10 in the [type] record for [param], but the PercentAvailability is greater than or equal to 90. When the PMA is greater than or equal to 90, you should only report an MODC of 10 to indicate that you used the maximum hourly value in the lookback period for the next available higher load bin, because there were no quality-assured data in the bin corresponding to the current load range. (See Part 75.33(c)(5).)	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2 Mass Rate Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ CO2M Derived Hourly Evaluation (LME)
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ Heat Input Derived Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ HIT Derived Hourly Evaluation (LME)
7	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Derived Hourly Evaluation
9	Process/Category:	Emissions Data Evaluation Report ------ NOXM Derived Hourly Evaluation (LME)
10	Process/Category:	Emissions Data Evaluation Report ------ SO2 Derived Hourly Evaluation
11	Process/Category:	Emissions Data Evaluation Report ------ SO2M Derived Hourly Evaluation (LME)
12	Process/Category:	Emissions Data Evaluation Report ------ SO2R Derived Hourly Evaluation

Check Code: HOURDHV-15

Check Name: Check Prior QA'd Hours for MODC 07

Related Former Checks:

Applicability: CEM Check

Description: For Method of Determination Code 07, all prior hours in reporting period are checked to ensure that total of QA'd hours is below a certain threshold

Specifications:

if (***Derived Hourly Modc Status*** == true AND ***Derived Hourly PMA Status*** == true)
    if (***Current DHV Record***.ModcCode == 07)

        case (***Current DHV Parameter***)

            NOXR: *MODC Set* = {01, 02, 04, 14, 21, 22, 53}
            CO2C: *MODC Set* = {01, 02, 04, 21, 53}
            H2O: *MODC Set* = {01, 02, 04, 21, 53}

        Prior QA Hours = count DerivedHourlyValueData records where
            DerivedHourlyValueData.ModcCode in set *MODC Set* AND
            DerivedHourlyValueData.ParameterCode = ***Current DHV Record***.ParameterCode AND
            (DerivedHourlyValueData.BeginDate < ***Current Date*** OR
              (DerivedHourlyValueData.BeginDate = ***Current Date*** AND DerivedHourlyValueData.BeginHour < ***Current Hour***))

        if (***Current DHV Parameter*** == "NOXR")
            if (*Prior QA Hours* > 2160)
                Derived Hourly Missing Data Status = false
                return result A
        else
            if (*Prior QA Hours* > 720)
                Derived Hourly Missing Data Status = false
                return result A

Results:

Result	Response	Severity
A	You reported an MODCCode of 07 in the DHV record for [param], but too many prior quality assured hours exist in evaluation period for use of this missing data approach.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation

Check Code: HOURDHV-16

Check Name: Check for Correct Use of Missing Data MODCs

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

Current DHV HBHA Value = null

if (***Derived Hourly Modc Status*** == true AND ***Derived Hourly PMA Status*** == true)

case (***Current DHV Parameter***)

NOXR: *MODC Set* = {01, 02, 03, 04, 14, 21, 22, 53, 54}
CO2C: *MODC Set* = {01, 02, 03, 04, 21, 53, 54}
H2O: *MODC Set* = {01, 02, 03, 04, 21, 53, 54}

if (***Current DHV Record***.ModcCode in set {06, 08, 09})

If (***Current DHV Parameter*** in set {CO2C, H2O})

Prior Record = latest DerivedHourlyValueData record or MonitorHourlyValueData record where
ParameterCode = ***Current DHV Parameter*** AND
ModcCode in set *MODC Set* AND
(Date < ***Current Date*** OR
(Date = ***Current Date*** AND Hour < ***Current Hour***))

If *Prior Record* is not null and is in current reporting period

Next Record = earliest DerivedHourlyValueData record or MonitorHourlyValueData record where
Data.ParameterCode = ***Current MHV Parameter*** AND
Data.ModcCode in set *MODC Set* AND
(Date > ***Current Date*** OR
(Date = ***Current Date*** AND Hour > ***Current Hour***))

If *Next Record* is not null and is in current reporting period

If *Prior Record*.AdjustedHourlyValue >= 0 AND *Next Record*.AdjustedHourlyValue >= 0

Current DHV HBHA Value = (*Prior Record*.AdjustedHourlyValue + *Next Record*.AdjustedHourlyValue) / 2, ROUNDED to a single decimal.

else

Derived Hourly Missing Data Status = false
return result A

else

If (***Current DHV Parameter*** == "NOXR") AND (***Primary Bypass Active For Hour*** == true)
PrimaryOrPrimaryBypassSystemKey = ***Current DHV Parameter***.SystemKey
else
PrimaryOrPrimaryBypassSystemKey = null

Prior DHV Record = latest DerivedHourlyValueData record where
(*PrimaryOrPrimaryBypassSystemKey* == null OR DerivedHourlyValueData.SystemKey== *PrimaryOrPrimaryBypassSystemKey*)

DerivedHourlyValueData.ParameterCode = ***Current DHV Parameter*** AND
DerivedHourValueData.ModcCode in set *MODC Set* AND
[DerivedHourlyValueData.Date < ***Current Date*** OR
(DerivedHourlyValueData.Date = ***Current Date*** AND DerivedHourlyValueData.Hour < ***Current Hour***)]

If *Prior DHV Record* is not null and is in current reporting period

Next DHV Record = earliest DerivedHourlyValueData record where
(*PrimaryOrPrimaryBypassSystemKey* == null OR DerivedHourlyValueData.SystemKey== *PrimaryOrPrimaryBypassSystemKey*)
DerivedHourlyValueData.ParameterCode = ***Current MHV Parameter*** AND
DerivedHourValueData.ModcCode in set *MODC Set* AND
[DerivedHourlyValueData.Date > ***Current Date*** OR
(DerivedHourlyValueData.Date = ***Current Date*** AND DerivedHourlyValueData.Hour > ***Current Hour***)]

If *Next DHV Record* is not null and is in current reporting period

If *Prior DHV Record*.AdjustedHourlyValue >= 0 AND *Next DHV Record*.AdjustedHourlyValue >= 0

Current DHV HBHA Value = (*Prior DHV Record*.AdjustedHourlyValue + *Next DHV Record*.AdjustedHourlyValue) / 2, ROUNDED to three decimal places.

else
Derived Hourly Missing Data Status = false
return result A

else if (***Current DHV Record***.ModcCode == "11")

Prior Measured DHV Record = DerivedHourlyValueData record at latest time for the location where
DerivedHourlyValueData.ModcCode in set *MODC Set* AND
DerivedHourlyValueData.ParameterCode = ***Current DHV Parameter*** AND
(DerivedHourlyValueData.BeginDate < ***Current Date*** OR
(DerivedHourlyValueData.BeginDate = ***Current Date*** AND DerivedHourlyValueData.BeginHour < ***Current Hour***))

If *Prior Measured DHV Record* is not null and is in the current reporting period
PriorDate = *Prior Measured DHV Record*.BeginDate
PriorHour = *Prior Measured DHV Record*.BeginHour
else
PriorDate = the day prior to the beginning of the current reporting period
PriorHour = 23

Next Measured DHV Record = DerivedHourlyValueData record at earliest time for the location where
DerivedHourlyValueData.ModcCode in set *MODC Set* AND
DerivedHourlyValueData.ParameterCode = ***Current DHV Parameter*** AND
(DerivedHourlyValueData.BeginDate > ***Current Date*** OR
(DerivedHourlyValueData.BeginDate = ***Current Date*** AND DerivedHourlyValueData.BeginHour > ***Current Hour***))

If *Next Measured DHV Record* is not null and is in the current reporting period
NextDate = *Next Measured DHV Record*.BeginDate
NextHour = *Next Measured DHV Record*.BeginHour
else
NextDate = the day after the end of the current reporting period

NextHour = 0

Missing Data Period Length = Count of DerivedHourlyValueData records for the location where
DerivedHourlyValueDataParameterCode = ***Current DHV Parameter*** AND
(DerivedHourlyValueData.BeginDate > *PriorDate* OR
(DerivedHourlyValueData.BeginDate = *PriorDate* AND DerivedHourlyValueData.BeginHour > *PriorHour*)) AND
(DerivedHourlyValueData.BeginDate < *NextDate* OR
(DerivedHourlyValueData.BeginDate = *NextDate* AND DerivedHourlyValueData.BeginHour < *NextHour*))

if (***Current DHV Record.***PercentAvailable is null OR ***Current DHV Record.***PercentAvailable >= 95.0)
if (*Missing Data Period Length* > 24)
Derived Hourly Missing Data Status = false
return result B
else
if (*Missing Data Period Length* > 8)
Derived Hourly Missing Data Status = false
return result B

Results:

Result	Response	Severity
A	The AdjustedHourlyValue reported in the DHV record for [param] either before or after the current hour is invalid.	Critical Error Level 1
B	You reported an MODCCode of 11 in the DHV record for [param], but the length of the missing data period exceeds the allowable value for use of this missing data procedure.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation

Check Code: HOURDHV-17

Check Name: Check Extraneous Data in DHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that certain fields are null in the DHV record.

Specifications:

Derived Hourly Null Status = false
Hourly Extraneous Fields = null

if (***Current DHV Record***.UnadjustedHourlyValue is not null)
    append "UnadjustedHourlyValue" to ***Hourly Extraneous Fields***

if (***Current DHV Record***.SegmentNumber is not null)
    append "SegmentNumber" to ***Hourly Extraneous Fields***

if (***Current DHV Record***.OperatingConditionCode is not null)
    if (***Current DHV Parameter*** is not equal to "NOXM")
        append "OperatingConditionCode" to ***Hourly Extraneous Fields***

if (***Current DHV Record***.FuelCode is not null)
    if (***Current DHV Parameter*** not in set {NOXM,SO2M,CO2M})
        append "FuelCode" to ***Hourly Extraneous Fields***

if (***Hourly Extraneous Fields*** is not null)
    return result A
else
    Derived Hourly Null Status = true

Results:

Result	Response	Severity
A	You reported [fieldnames] in the DHV record for [param]. This data should be blank.	Non-Critical Error

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2 Mass Rate Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ CO2M Derived Hourly Evaluation (LME)
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ Heat Input Derived Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ HIT Derived Hourly Evaluation (LME)
7	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Derived Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ NOXM Derived Hourly Evaluation (LME)
9	Process/Category:	Emissions Data Evaluation Report ------ SO2 Derived Hourly Evaluation
10	Process/Category:	Emissions Data Evaluation Report ------ SO2M Derived Hourly Evaluation (LME)
11	Process/Category:	Emissions Data Evaluation Report ------ SO2R Derived Hourly Evaluation

Check Code: HOURDHV-18

Check Name: Check System in DHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that a valid Monitoring System is indicated in the DHV record.

Specifications:

Current DHV Mon Sys Record = null
Derived Hourly System Status = false

if (***Current DHV Parameter*** == "NOXR")
    App E Constant Fuel Mix = false

If ((***Current DHV Parameter*** in set {SO2, SO2R, NOX, CO2} AND ***Current DHV Method*** <> "AMS") OR
    (***Current DHV Parameter*** == "HI" AND ***Current DHV Method*** in set {CALC, AD, ADCALC}) OR
    (***Current DHV Parameter*** == "H2O" AND ***Current DHV Method*** == "MDF") OR
    LME HI Method is not null)

    if ***Current DHV Record***.MonitoringSystemID is NOT null
        return result A
    else
        Derived Hourly System Status = true

Else If (***Current DHV Parameter*** <> "HI")

    case (***Current DHV Parameter***)

        NOXR: *MODC Set* = {01, 02, 03, 04, 14, 21, 22}
        CO2C: *MODC Set* = {01, 02, 03, 04, 21}
        H2O: *MODC Set* = {01, 02, 03, 04, 21}

    If (***Current DHV Record***.MonitoringSystemID is null)

        If (***Current DHV Method*** == "AMS")
            Derived Hourly System Status = true

        Else If (***Current DHV Method*** == "AE")
            If ***Current DHV Record***.OperatingConditionCode is null
                Derived Hourly System Status = true
            else
                return result J

        Else If (***Current DHV Record.***ModcCode in set *MODC Set*)
            return result C
        Else If (***Current DHV Parameter*** == "CO2C" AND ***Current DHV Method*** == "CEM")
            return result K
        Else If (***Current DHV Parameter*** == "NOXR" AND ***Current DHV Method*** == "CEM" AND ***Current DHV Record.***ModcCode <> "23")
            return result K
        Else If (***Current DHV Parameter*** == "H2O" AND ***Current DHV Method*** in set { "MMS", "MTB", "MWD" })
            return result K
        Else
            Derived Hourly System Status = true

    else

```
If ( Derived Hourly MODC Status == true AND Current DHV Method in set {CEM, PEM, MWD} AND Current DHV
Record.ModcCode not in set {05, 53, 54} AND Current DHV Record.ModcCode not in set MODC Set
    AND NOT ( Current DHV Parameter in set { "CO2C", "NOXR" } AND Current DHV Method == "CEM" )
    AND NOT ( Current DHV Parameter == "H2O" AND Current DHV Method ==  "MWD" ) )
        return result B

else
        Current DHV Mon Sys Record = find active MonitoringSystemData record for location where
        MonitoringSystemData.MonitoringSystemID = Current DHV Record.MonitoringSystemID

        if (Current DHV Mon Sys Record is null)
                return result D

        else if (Derived MHV Mon Sys Record.SystemTypeCode <> Current DHV System Type)
                return result E

        else if (Current DHV Method == "AE" AND Hourly Fuel Flow Count for Gas + Hourly Fuel Flow Count for
        Oil > 0)

                If (Derived DHV Mon Sys Record.FuelCode == "MIX" OR Current DHV
                Record.OperatingConditionCode is not null)
                        if (Current DHV Record.OperatingConditionCode == "E")
                                return result F
                        else
                                App E Constant Fuel Mix = true
                                App E Reporting Method = "CONSTANT"
                                App E Reported Value = Current DHV Record.AdjustedHourlyValue
                                App E Segment Number = Current DHV Record.SegmentNumber
                                App E Fuel Code = "MIX"
                                App E NOXE System ID  = Current DHV Record.MonitoringSystemID
                                App E NOXE System Identifier  = Current DHV Record.SystemIdentifier
                                Derived Hourly System Status = true
                                if (Current DHV Record.OperatingConditionCode in set {X, Y, Z, U, W, N, M})
                                        App E Op Code = Current DHV Record.OperatingConditionCode
                                        if (Derived DHV Mon Sys Record.FuelCode <> "MIX")
                                                return result G
                                else
                                        return result H
                        else
                                return result I
        else
                Derived Hourly System Status = true
```

Results:

Result	Response	Severity
A	You reported [fieldnames] in the DHV record for [param]. This data should be blank.	Non-Critical Error
B	You reported a MonitoringSystemID in the DHV record for [param]. This field should be blank when missing data substitution is used.	Non-Critical Error
C	You did not report a MonitoringSystemID in the DHV record for [param]. This information is required when you report measured data.	Critical Error Level 1
D	You reported MonitoringSystemID [ID] in the DHV record for [param], but according to your monitoring plan this system was not active during the hour.	Critical Error Level 1
E	You reported MonitoringSystemID [ID] in the DHV record for [param], but the SystemTypeCode of this system is not appropriate.	Critical Error Level 1
F	You reported an OperatingConditionCode of E in the DHV record for NOXR. You should report the NOx emission rate for emergency fuels in an HPFF record, not a DHV record.	Critical Error Level 1
G	You reported an OperatingConditionCode in the DHV record for NOXR, which indicates that you are determining NOx emission rate from a mixed fuel Appendix E curve, but the FuelCode of NOXE MonitoringSystemID [ID] is not equal to "MIX". If a NOXE system measures an individual fuel, the emissions from this system should be reported in an HPFF record, not a DHV record.	Critical Error Level 1
H	The OperatingConditionCode reported in the DHV record for NOXR is missing or invalid.	Critical Error Level 1
I	You reported NOXE MonitoringSystemID [ID] in the DHV record for NOXR, but the FuelCode of this system is not equal to "MIX". If a NOXE system measures an individual fuel, the emissions from this system should be reported in an HPFF record. If this data represents unit-level emissions based on fuel-specific emissions data that have been reported in one or more HPFF records, then the MonitoringSystemID should be blank.	Critical Error Level 1
J	You reported an OperatingConditionCode in the DHV record for NOXR, which indicates that you are determining NOx emission rate using Appendix E, but you did not report a MonitoringSystemID in this record. If you determined the NOx emission rate from a mixed fuel curve or via heat input apportionment, you should report the MonitoringSystemID of the NOXE system for the curve. If you determined the NOx emission rate from one or more individual fuel curves, you should not report an OperatingConditionCode in the NOXR DHV record.	Critical Error Level 1
K	You did not report a MonitoringSystemID in the DHV record for [param]. This information is required when you report missing data.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2 Mass Rate Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ CO2M Derived Hourly Evaluation (LME)
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ Heat Input Derived Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ HIT Derived Hourly Evaluation (LME)
7	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Derived Hourly Evaluation
9	Process/Category:	Emissions Data Evaluation Report ------ NOXM Derived Hourly Evaluation (LME)
10	Process/Category:	Emissions Data Evaluation Report ------ SO2 Derived Hourly Evaluation
11	Process/Category:	Emissions Data Evaluation Report ------ SO2M Derived Hourly Evaluation (LME)
12	Process/Category:	Emissions Data Evaluation Report ------ SO2R Derived Hourly Evaluation

Check Code: HOURDHV-19

Check Name: Check System Designation Code for System in DHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that the SystemDesignationCode of the monitoring system is compatible with reported MODC.

Specifications:

If (***Derived Hourly Modc Status*** == true AND ***Derived Hourly System Status*** == true AND ***Current DHV Mon Sys Record*** is not null)

case (***Current DHV Record.***ModcCode)

01: If (***Current DHV Mon Sys Record***. SystemDesignationCode NOT in set {P, PB})
return result A

02: If (***Current DHV Mon Sys Record***. SystemDesignationCode NOT in set {B, RB, DB}
return result B

04: If (***Current DHV Mon Sys Record***. SystemDesignationCode <> "RM")
return result C

22: If (***Current DHV Mon Sys Record***. SystemDesignationCode <> "CI")
return result D

Results:

Result	Response	Severity
A	You reported an MODCCode of [modcCode] in the DHV record for [param], but MonitoringSystemID [ID] is not a primary system.	Critical Error Level 1
B	You reported an MODCCode of [modcCode] in the DHV record for [param], but MonitoringSystemID [ID] is not a backup system.	Critical Error Level 1
C	You reported an MODCCode of [modcCode] in the DHV record for [param], but MonitoringSystemID [ID] is not a reference method system.	Critical Error Level 1
D	You reported an MODCCode of 22 in the DHV record for NOXR, but MonitoringSystemID [ID] is not a certified inlet system.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation

Check Code: HOURDHV-24

Check Name: Check Formula in DHV Record

Related Former Checks:

Applicability: CEM Check

Description: Checks the Formula ID in the DerivedHourlyValue record and ensures that it can be used for the calculation.

Specifications:

Derived Hourly Formula Status = false
Derived Hourly Equation Status = false
Current DHV Multiple Fuel Equation Code == null
Current DHV Formula Record = null

If (***Current DHV Record***.FormulaIDKey is null)
    If (***Current DHV Method*** in set {AMS, LME}) OR
        (***Derived Hourly Modc Status*** == true AND ***Current DHV Record***.ModcCode == "40") OR
        LME HI Method is not null)
        Derived Hourly Formula Status = true

    else if (***Current DHV Parameter*** = "AE" AND ***App E Constant Fuel Mix*** == true)
        Derived Hourly Formula Status = true

    else if (***Current DHV Parameter*** in set {NOXR, SO2, HI, CO2}) AND ***Current DHV Method*** in set {AD, AE})
        Derived Hourly Formula Status = true

        If (***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil*** > 1)

            case (***Current DHV Parameter***)
                NOXR: ***Current DHV Multiple Fuel Equation Code*** = "E-2"
                SO2: ***Current DHV Multiple Fuel Equation Code*** = "D-12"
                CO2: ***Current DHV Multiple Fuel Equation Code*** = "G-4A"
                HI: ***Current DHV Multiple Fuel Equation Code*** = "D-15A"

            Locate active Formula Record for location WHERE
                ParameterCode == ***Current DHV Parameter*** AND
                EquationCode == ***Current DHV Multiple Fuel Equation Code***

            If found,
                If ***(Legacy Data Evaluation*** == true)
                    return result A
                else
                    return result B

    else if (***Current DHV Method*** = "PEM")
        Derived Hourly Formula Status = true

    else if (***Current DHV Parameter*** == "NOX" AND ***Current NOx Rate Method Code*** == "AE" AND ***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil*** > 1 AND ***Legacy Data Evaluation*** == true)
        Derived Hourly Formula Status = true

    else if (***Current DHV Parameter*** in set {NOXR, H2O, CO2C})
        If (***Derived Hourly Modc Status*** == true)
            If (***Current DHV Record***.ModcCode in set {01, 02, 03, 04, 05, 14, 21, 22, 53, 54})
                return result C
            else
                Derived Hourly Formula Status = true
                return result K

```
        else
                return result C

else
        If (Current DHV Parameter in set {SO2R, H2O} AND Current DHV Record.MODCCode == "40")
                return result D

        else if (LME HI Method is not null)
                return result J
        else

                Current DHV Formula Record= Find MonitoringFormulaData record where
                        MonitoringFormulaData,MonitoringFormulaIDKey = Current DHV Record.FormulaIDKey

                If (Current DHV Formula Record is null)
                        return result E

                else if (Current DHV Formula Record.ParameterCode is not equal to Current DHV Parameter)
                        If Current DHV Parameter == "HI" AND Current DHV Method = "AD" AND Hourly Fuel Flow Count for
                        Gas + Hourly Fuel Flow Count for Oil > 1 AND Current DHV Formula Record.ParameterCode == "HIT"
                        AND Current DHV Formula Record.EquationCode == "D-15" AND Legacy Data Evaluation == true)
                                return result I
                        else
                                return result F

                else
                        Derived Hourly Formula Status = true

                        if Current DHV Parameter == "HI" AND Current DHV Method = "ADCALC" and Current DHV Formula
                        Record.EquationCode not in {F-21A,F-21B,F-21D}

                                Count all active MonitoringFormulaData record for location where
                                        EquationCode in {F-21A,F-21B,F-21D}

                                if (Count = 1)
                                        Current DHV Formula Record = Find active MonitoringFormulaData record for location where
                                                EquationCode in {F-21A,F-21B,F-21D}

                        else if (Current DHV Method == "AE")
                                if  (App E Constant Fuel Mix == true OR Hourly Fuel Flow Count for Gas + Hourly Fuel Flow Count
                                for Oil == 0)
                                        return result H
```

Results:

Result	Response	Severity
A	You did not report a FormulaID in the DHV record for [param]. While this is acceptable for legacy EDR data, the FormulaID will be required for ECMPS.	Informational Message
B	You did not report a FormulaID in the DHV record for [param]. This formula is required when you burn multiple fuels during the hour.	Critical Error Level 1
C	You did not report a FormulaID in the DHV record for [param].	Critical Error Level 1
D	You reported an MODC of 40, but you reported a FormulaID in the DHV record for [param]. This field should be blank when reporting a default value.	Critical Error Level 1
E	You reported FormulaID [ID] in the DHV record for [param], but there is no active Formula record for this formula in your monitoring plan.	Critical Error Level 1
F	You reported FormulaID [ID] in the DHV record for [param], but this is not a [param] formula.	Critical Error Level 1
G	(Obsolete) You reported a FormulaID in the DHV record for [param]. This field should be blank when using missing data substitution.	Non-Critical Error
H	You reported a FormulaID in the DHV record for NOXR. This field should be blank unless you determine the NOX emission rate using multiple Appendix E curves.	Critical Error Level 1
I	You reported FormulaID [ID] in the DHV record for HI, but FormulaCode D-15 will no longer be appropriate for calculating HI from multiple fuels. For ECMPS, the ParameterCode should be for this formula should be HI and the FormulaCode should be "D-15A".	Informational Message
J	You reported a FormulaID in the DHV record for [param]. This field should be blank when reporting emissions for LME units.	Critical Error Level 1
K	You did not report a FormulaID in the DHV record for [param].	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2 Mass Rate Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ CO2M Derived Hourly Evaluation (LME)
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ Heat Input Derived Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ HIT Derived Hourly Evaluation (LME)
7	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Derived Hourly Evaluation
9	Process/Category:	Emissions Data Evaluation Report ------ NOXM Derived Hourly Evaluation (LME)
10	Process/Category:	Emissions Data Evaluation Report ------ SO2 Derived Hourly Evaluation
11	Process/Category:	Emissions Data Evaluation Report ------ SO2M Derived Hourly Evaluation (LME)
12	Process/Category:	Emissions Data Evaluation Report ------ SO2R Derived Hourly Evaluation

Check Code: HOURDHV-25

Check Name: Check Heat Input Equation Code

Related Former Checks:

Applicability: General Check

Description: Looks up the equation code for the current Heat Input Derived Hourly Record and verifies that it is appropriate for heat input calculations.

Specifications:

CO2 Conc Checks Needed for Heat Input = false
O2 Wet Checks Needed for Heat Input = false
O2 Dry Checks Needed for Heat Input = false

if (***Heat Input App D Method Active For Hour*** == true)
    Hourly Fuel Flow Checks needed for Heat Input = true
else
    Hourly Fuel Flow Checks needed for Heat Input = false

Heat Input Equation Code = null
if (***Derived Hourly Formula Status*** == true)
    if (***Current DHV Formula Record*** is not null)
        Heat Input Equation Code = ***Current DHV Formula Record***.EquationCode
        if (***Heat Input CEM Method Active For Hour*** == true)
            if (***Heat Input Equation Code*** in set {F-15, F-16, F-17, F-18})
                Derived Hourly Equation Status = true
                Flow Monitor Hourly Checks Needed = true
                Flow Needed For Part 75 = true

                if (***Heat Input Equation Code*** <> "F-15")
                    Moisture Needed = true
                    append "MIN" to ***H2O Missing Data Approach***

                if (***Heat Input Equation Code*** = "F-15" OR ***Heat Input Equation Code*** = "F-16")
                    CO2 Conc Checks Needed for Heat Input = true
                    FC Factor Needed = true
                else if (***Heat Input Equation Code*** = "F-17")
                    O2 Wet Checks Needed for Heat Input = true
                    FD Factor Needed = true
                else if (***Heat Input Equation Code*** = "F-18")
                    O2 Dry Checks Needed for Heat Input = true
                    FD Factor Needed = true
            else if (***Heat Input Equation Code*** is null)
                return result A
            else
                return result B
        else if (***Heat Input App D Method Active For Hour*** == true)
            if (***Heat Input Equation Code*** == "D-15A")
                Derived Hourly Equation Status = true
            else if (***Heat Input Method Code*** == "ADCALC" and ***Heat Input Equation Code*** in set {F-21A, F-21B, F-21C, F-21D, F-25})
                Derived Hourly Equation Status = true
                if (***Heat Input Equation Code*** == "F-21D")
                    Apportionment HI Method Array for the location = "NOCALC"
            else if (***Heat Input Equation Code*** in set {F-19, F-19V, F-20, D-6, D-8} AND (***Legacy Data Evaluation*** == true OR ***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil*** == 1))
                Derived Hourly Equation Status = true
                return result C

```
                else if (Heat Input Equation Code is null)
                        return result A
                else
                        return result B
        else if (Current DHV Method in set {CALC, ADCALC})
                if (Heat Input Equation Code in set  {F-21A, F-21B, F-21C, F-25})
                        Derived Hourly Equation Status = true
                else if (Heat Input Equation Code == "SS-3B")
                        Derived Hourly Equation Status = true
                        Apportionment HI Method Array for the location = "COMPLEX"
                else if (Heat Input Equation Code == "F-21D" OR Current DHV Method == "ADCALC")
                        Derived Hourly Equation Status = true
                        Apportionment HI Method Array for the location = "NOCALC"
                else if (Heat Input Equation Code is null)
                        return result A
                else
                        return result B
        else
                Derived Hourly Equation Status = true
else
        Derived Hourly Equation Status = true
```

Results:

Result	Response	Severity
A	You reported FormulaID [ID] in the DHV record for [param], but you did not report a FormulaCode for this formula in your monitoring plan.	Critical Error Level 1
B	You reported FormulaID [ID] in the DHV record for HI, but the FormulaCode of this formula is not appropriate for calculating HI.	Critical Error Level 1
C	You reported FormulaID [ID] in the DHV record for HI, but a formula with a FormulaCode [EQCODE] is no longer appropriate in this record. For ECMPS, if you are calculating heat input from multiple fuels using Appendix D, you should report a formula with a FormulaCode of D-15A in the DHV record; otherwise, do not report a FormulaID.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Heat Input Derived Hourly Evaluation

Check Code: HOURDHV-26

Check Name: Check NOX Equation Code

Related Former Checks:

Applicability: CEM Check

Description: Retrieves and validates NOx Mass Equation Code as a valid formula code for calculating NOx Mass

Specifications:

NOx Rate Checks Needed for NOx Mass Calc = false
Heat Input Checks Needed for NOx Mass Calc = false

NOx Mass Equation Code = null
if (***Derived Hourly Formula Status*** == true)
    if (***Current DHV Formula Record*** is not null)
        NOx Mass Equation Code = ***Current DHV Formula Record***.EquationCode
        if (***Current DHV Method*** == "CEM")
            if (***NOx Mass Equation Code*** in set {F-26A, F-26B})
                Derived Hourly Equation Status = true
                Flow Monitor Hourly Checks Needed = true
                Flow Needed For Part 75 = true

                if (***NOx Mass Equation Code*** = "F-26B") // note that the old name for this formula was "N-2"
                    Moisture Needed = true
                    append "MIN" to ***H2O Missing Data Approach***

            else if (***NOx Mass Equation Code*** is null)
                return result A
            else
                return result B

        else if (***Current DHV Method*** == "NOXR")
            if (***NOx Mass Equation Code*** == "F-24A")
                Derived Hourly Equation Status = true
                Heat Input Checks Needed for NOx Mass Calc = true
                Nox Rate Checks Needed for NOx Mass Calc = true

            else if (***NOx Mass Equation Code*** is null)
                return result A
            else
                return result C
        else
            Derived Hourly Equation Status = true
    else
        Derived Hourly Equation Status = true
        if (***Current NOx Rate Method Code*** == "AE" AND ***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil*** > 1 AND ***Legacy Data Evaluation*** == true)
            NOx Mass Equation Code = "F-24A"

Results:

Result	Response	Severity
A	You reported FormulaID [ID] in the DHV record for [param], but you did not report a FormulaCode for this formula in your monitoring plan.	Critical Error Level 1
B	You reported FormulaID [ID] in the DHV record for NOX, but the FormulaCode of this formula is not appropriate for calculating NOX from a NOXC system.	Critical Error Level 1
C	You reported FormulaID [ID] in the DHV record for NOX, but the FormulaCode of this formula is not appropriate for calculating NOX from a NOx-diluent system.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Derived Hourly Evaluation

Check Code: HOURDHV-27

Check Name: Check NOXR Equation Code

Related Former Checks:

Applicability: CEM Check

Description: Gets Equation Code from Active Monitor Formula Record and verifies that it is an appropriate equation for calculation of NOx Rate.

Specifications:

O2 Dry Checks Needed for NOx Rate Calc = false
O2 Wet Checks Needed for NOx Rate Calc = false
CO2 Diluent Checks Needed for NOx Rate Calc = false

if (***Current DHV Method*** == "AE")
    Hourly Fuel Flow Needed for NOx Rate Calc = true
else
    Hourly Fuel Flow Needed for NOx Rate Calc = false

NOx Rate Equation Code = "" // null string
if (***Derived Hourly Formula Status*** == true)
    if (***Current DHV Formula Record*** is not null)
        NOx Rate Equation Code = ***Current DHV Formula Record***.EquationCode

        if (***Current DHV Method*** == "CEM")
            if (***NOx Rate Equation Code*** in set {19-1, 19-2, 19-3, 19-3D, 19-4, 19-5, 19-5D, 19-6, 19-7, 19-8, 19-9, F-5, F-6})
                Derived Hourly Equation Status = true

                If (***Current DHV Record***.ModcCode != "23")

                    If (***NOx Rate Equation Code*** in set {19-1, 19-4, F-5})

                        O2 Dry Checks Needed for NOx Rate Calc = true

                        If (***Current DHV Record***.ModcCode in set {01, 02, 03, 04, 05, 14, 21, 22, 53, 54})
                            FD Factor Needed = true

                    else if (***NOx Rate Equation Code*** in set {19-3, 19-5})

                        O2 Wet Checks Needed for NOx Rate Calc = true

                        If (***Current DHV Record***.ModcCode in set {01, 02, 03, 04, 05, 14, 21, 22, 53, 54})
                            FD Factor Needed = true

                    else if (***NOx Rate Equation Code*** in set {19-3D, 19-5D})

                        If (***Current DHV Record***.ModcCode in set {01, 02, 03, 04, 05, 14, 21, 22, 53, 54})
                            FD Factor Needed = true

                    else if (***NOx Rate Equation Code*** in set {19-6, 19-7, 19-8, 19-9, F-6})

                        CO2 Diluent Checks Needed for NOx Rate Calc = true

                        If (***Current DHV Record***.ModcCode in set {01, 02, 03, 04, 05, 14, 21, 22, 53, 54})
                            FC Factor Needed = true

```
                                    else if (NOx Rate Equation Code == "19-2" )

                                            O2 Wet Checks Needed for NOx Rate Calc = true

                                            If (Current DHV Record.ModcCode in set {01, 02, 03, 04, 05, 14, 21, 22, 53, 54})
                                                    FW Factor Needed = true

                                    if (NOx Rate Equation Code in set {19-3, 19-3D, 19-4, 19-8}

                                            Moisture Needed = true
                                            append "MAX" to H2O Missing Data Approach

                                    else if (NOx Rate Equation Code in set {19-5, 19-9}
                                            Moisture Needed = true
                                            append "MIN" to H2O Missing Data Approach

                            else (if (NOx Rate Equation Code is null)
                                    return result A
                            else
                                    return result B

                    else if (Current DHV Method == "AE")
                            if (NOx Rate Equation Code == 'E-2')
                                    Derived Hourly Equation Status = true
                            else if (NOx Rate Equation Code is null)
                                    return result A
                            else
                                    return result C
                    else
                            Derived Hourly Equation Status = true
            else
                    Derived Hourly Equation Status = true
```

Results:

Result	Response	Severity
A	You reported FormulaID [ID] in the DHV record for [param], but you did not report a FormulaCode for this formula in your monitoring plan.	Critical Error Level 1
B	You reported FormulaID [ID] in the DHV record for NOXR, but the FormulaCode of this formula is not appropriate for calculating NOXR.	Critical Error Level 1
C	You reported FormulaID [ID] in the DHV record for NOXR, but the FormulaCode of this formula is not appropriate for calculating NOXR from multiple fuels. The FormulaCode should be E-2.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation

Check Code: HOURDHV-28

Check Name: Check CO2C Equation Code

Related Former Checks:

Applicability: CEM Check

Description: Gets Equation Code from Active Monitor Formula Record and verifies that it is an appropriate equation for calculation of CO2 Conc (Either F-14A or F-14B)

Specifications:

CO2 Conc CEM Equation Code = "" // null string
if (***Derived Hourly Formula Status*** == true)
    if (***Current DHV Formula Record*** is not null)
        CO2 Conc CEM Equation Code = ***Current DHV Formula Record***.FormulaCode
        if (***CO2 Conc CEM Equation Code*** in set {F-14A, F-14B})
            Derived Hourly Equation Status = true
        else
            return result A
    else
        Derived Hourly Equation Status = true

Results:

Result	Response	Severity
A	You reported FormulaID [ID] in the DHV record for CO2C, but the FormulaCode of this formula is not appropriate for calculating CO2C.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Derived Hourly Evaluation

Check Code: HOURDHV-29

Check Name: Check CO2 Equation Code

Related Former Checks:

Applicability: General Check

Description: Gets Equation Code from Active Monitor Formula Record and verifies that it is an appropriate equation for calculation of CO2 Mass (Either F-2 or F-11)

Specifications:

CO2 Conc Checks Needed for CO2 Mass Calc = false
Use CO2 Diluent Cap for Co2 Mass Calc = false
Use O2 Diluent Cap for Co2 Conc Calc = false
if (***CO2 App D Method Active For Hour*** == true)
    Hourly Fuel Flow Checks Needed for CO2= true
else
    Hourly Fuel Flow Checks Needed for CO2 = false

CO2 Mass Equation Code = "" // null string
if (***Derived Hourly Formula Status*** == true)

    if (***Current DHV Formula Record*** is not null)
        CO2 Mass Equation Code = ***Current DHV Formula Record***.FormulaCode
        if ***(CO2 CEM Method Active For Hour*** == true)
            Flow Monitor Hourly Checks Needed = true
            Flow Needed For Part 75 = true
            CO2 Conc Checks Needed for CO2 Mass Calc = true

            if (***CO2 Mass Equation Code***== "F-2" OR ***CO2 Mass Equation Code*** == "F-11")
                Derived Hourly Equation Status= true

                If (***CO2 Mass Equation Code***== 'F-2')
                    Moisture Needed = true
                    append "MIN" to ***H2O Missing Data Approach***

                if (***Current DHV Record***.DiluentCapIndicator == 1)
                    Use CO2 Diluent Cap for Co2 Mass Calc = true
                    Use O2 Diluent Cap for Co2 Conc Calc = true
            else
                return result A
        else if (***CO2 App D Method Active For Hour*** == true)
            if (***CO2 Mass Equation Code*** == "G-4A")
                Derived Hourly Equation Status= true

            else if (***CO2 Mass Equation Code***== "G-4" AND (***Legacy Data Evaluation*** == true OR ***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil*** == 1))
                Derived Hourly Equation Status= true
                return result B
            else
                return result A
        else
            Derived Hourly Equation Status = true
    else
        Derived Hourly Equation Status= true

Results:

Result	Response	Severity
A	You reported FormulaID [ID] in the DHV record for CO2, but the FormulaCode of this formula is not appropriate for calculating CO2.	Critical Error Level 1
B	You reported FormulaID [ID] in the DHV record for CO2, but a formula with a FormulaCode [EQCODE] is no longer appropriate in this record. For ECMPS, if you are calculating CO2 from multiple fuels using Appendix D, you should report a formula with a FormulaCode of G-4A in the DHV record; otherwise, do not report a FormulaID.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Mass Rate Derived Hourly Evaluation

Check Code: HOURDHV-30

Check Name: Check SO2 Equation Code

Related Former Checks:

Applicability: CEM Check

Description: Gets Equation Code from Active Monitor Formula Record and verifies that it is an appropriate equation for calculation of SO2 Mass (Either F-1 or F-2)

Specifications:

SO2 Monitor Hourly Checks Needed = false
if (***SO2 App D Method Active For Hour*** == true)
    Hourly Fuel Flow Checks Needed for SO2 = true
else
    Hourly Fuel Flow Checks Needed for SO2 = false

SO2 Equation Code = "" // null string
if (***Derived Hourly Formula Status*** == true)
    If (***Current DHV Formula Record*** is not null)
        SO2 Equation Code = ***Current DHV Formula Record***.FormulaCode

        if (***SO2 CEM Method Active For Hour*** == true)
            if (***SO2 Equation Code*** == "F-1" OR ***SO2 Equation Code*** == "F-2")
                Derived Hourly Equation Status = true
                Flow Monitor Hourly Checks Needed = true
                Flow Needed For Part 75 = true

                If (***SO2 Equation Code*** == "F-2")
                    Moisture Needed = true
                    append "MIN" to ***H2O Missing Data Approach***

                if (***SO2 Monitor Hourly Count*** == 0)
                    return result A
                else
                    SO2 Monitor Hourly Checks Needed = true

            else if (***SO2 Equation Code*** == "F-23" AND ***SO2 F23 Method Active For Hour*** == true)
                Derived Hourly Equation Status = true
            else
                return result B

        else if (***SO2 F23 Method Active For Hour*** == true)
            if (***SO2 Equation Code*** == "F-23")
                Derived Hourly Equation Status = true
            else
                return result B

        else if (***SO2 App D Method Active For Hour*** == true)
            if (***SO2 Equation Code*** = "D-12")
                Derived Hourly Equation Status = true
            else if (***SO2 Equation Code*** in {D-2, D-4, D-5} AND ***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil*** == 1))
                Derived Hourly Equation Status = true
                return result C
            else
                return result B
        else
            Derived Hourly Equation Status = true

else

Derived Hourly Equation Status = true

Results:

Result	Response	Severity
A	You did not report an MHV record for [param] for the hour.	Critical Error Level 1
B	You reported FormulaID [ID] in the DHV record for SO2, but the FormulaCode of this formula is not appropriate for calculating SO2.	Critical Error Level 1
C	You reported FormulaID [ID] in the DHV record for SO2, but a formula with a FormulaCode [EQCODE] is not appropriate in this record. If you are calculating SO2 from multiple fuels using Appendix D, you should report a formula with a FormulaCode of D-12 in the DHV record; otherwise, do not report a FormulaID.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ SO2 Derived Hourly Evaluation

Check Code: HOURDHV-31

Check Name: Check H2O Equation Code

Related Former Checks:

Applicability: CEM Check

Description: Looks up the Formula Identifier defined in the H20 Derived Hourly Record and ensures that it is a valid formula for H2O calculations

Specifications:

H2O CEM Equation Code = "" // null string
if (***Derived Hourly Formula Status*** == true)
    If (***Current DHV Formula Record*** is not null)
        H2O CEM Equation Code = ***Current DHV Formula Record***.FormulaCode
        if (***H2O CEM Equation Code*** in set (F-31, M-1K)
            Derived Hourly Equation Status = true
        else
            return result A
    else
        Derived Hourly Equation Status = true

Results:

Result	Response	Severity
A	You reported FormulaID [ID] in the DHV record for H2O, but the FormulaCode of this formula is not appropriate for calculating H2O.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation

Check Code: HOURDHV-32

Check Name: Verify Correct Reporting of NOXC MHV Record

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

If (***Current DHV Method*** in set {CEM, CEMNOXR})

If (***Current DHV Parameter*** == "NOXR")
Nox Conc Needed for NOx Rate Calc = false

If ***Derived Hourly MODC Status == true***
If (***NOx Conc Monitor Hourly Count*** == 0)
If ***Current DHV Record.***ModcCode in set {01, 02, 03, 04, 14, 21, 22, 53, 54}
return result A
Else If ***Current DHV Record.***ModcCode != "23"
return result C
else
Nox Conc Needed for NOx Rate Calc = true

Else if (***Current DHV Parameter*** == "NOX")
Nox Conc Needed for NOx Mass Calc = false

If (***Derived Hourly Equation Status*** == true AND ***NOx Mass Equation Code*** begins with "F-26")
if (***NOx Conc Monitor Hourly Count*** == 0)
return result A
else
Nox Conc Needed for NOx Mass Calc = true

Results:

Result	Response	Severity
A	You did not report an MHV record for NOXC for the hour.	Critical Error Level 1
B	(Obsolete) You reported an MHV record for NOXC, but you reported a MODCCode of [modc] in the DHV record for NOXR. You should not report an MHV record for NOXC when you use substitute data to determine the NOx emission rate.	Non-Critical Error
C	You did not report an MHV record for NOXC for the hour. An MHV record is required for missing data.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Derived Hourly Evaluation

Check Code: HOURDHV-33

Check Name: Determine Default Value for MODC 40

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

Derived Hourly Default Status == true
If (***Derived Hourly Modc Status*** == true AND ***Current DHV Record***.ModcCode == 40)

If (***Current DHV Parameter*** == "H2O")

If ***Current DHV Record***.AdjustedHourlyValue is null OR
Current DHV Record.AdjustedHourlyValue <= 0 OR
Current DHV Record.AdjustedHourlyValue >= 100
Derived Hourly Default Status == false
return result A

If ***H2O Default Max Value*** is null
If (***H2O Default Value*** > 0 AND ***H2O Default Value*** < 100)
if (***Current DHV Record***.AdjustedHourlyValue <> ***H2O Default Value***)
Derived Hourly Default Status == false
return result B

else if (***H2O Default Max Value*** > 0 AND ***H2O Default Max Value*** < 100 AND ***H2O Default Min Value*** > 0 AND ***H2O Default Min Value*** < 100)
If ***Current DHV Record***.AdjustedHourlyValue < ***H2O Default Min Value*** OR
Current DHV Record.AdjustedHourlyValue > ***H2O Default Max Value***)
Derived Hourly Default Status == false
return result C
Else
H2O Default Value = ***Current DHV Record***.AdjustedHourlyValue

Else if (***Current DHV Parameter*** == "SO2R")

If ***Current DHV Record***.AdjustedHourlyValue is null OR
Current DHV Record.AdjustedHourlyValue <= 0
Derived Hourly Default Status == false
return result D

If ***F23 Default Max Value*** is null
If (***F23 Default Value*** > 0)
if (***Current DHV Record***.AdjustedHourlyValue <> ***F23 Default Value***)
Derived Hourly Default Status == false
return result B

else if (***F23 Default Max Value*** > 0 AND ***F23 Default Min Value*** > 0)
If ***Current DHV Record***.AdjustedHourlyValue < ***F23 Default Min Value*** OR
Current DHV Record.AdjustedHourlyValue > ***F23 Default Max Value***)
Derived Hourly Default Status == false
return result C
Else
F23 Default Value = ***Current DHV Record***.AdjustedHourlyValue

Results:

Result	Response	Severity
A	The AdjustedHourlyValue reported in the DHV record for [param] is invalid. The value must be between 0 and 100.	Critical Error Level 1
B	You reported an MODCCode of 40 in the DHV record for [param], but the AdjustedHourlyValue is not equal to the active default value in your monitoring plan.	Critical Error Level 1
C	You reported an MODCCode of 40 in the DHV record for [param], but the AdjustedHourlyValue is outside the range of the active default values in your monitoring plan.	Critical Error Level 1
D	The AdjustedHourlyValue reported in the DHV record for [param] is invalid. The value must be greater than 0.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ SO2R Derived Hourly Evaluation

Check Code: HOURDHV-34

Check Name: Determine Derived Hourly Record Status

Related Former Checks:

Applicability: General Check

Description:

Specifications:

If (***Current DHV Parameter*** == "NOXR")
Current NOX System Status = Derived Hourly System Status
Current NOXR HBHA Value = Current DHV HBHA Value
else if (***Current DHV Parameter*** == "CO2C")
Current CO2C DHV HBHA Value = Current DHV HBHA Value
else if (***Current DHV Parameter*** == "H2O")
Current H2O DHV HBHA Value = Current DHV HBHA Value

If (***Derived Hourly Modc Status*** == false OR ***Derived Hourly Equation Status*** == false OR ***Derived Hourly Missing Data Status*** == false
OR (***Current DHV Record.***MODCCode in set {06, 07, 08, 09, 10, 11) AND ***Derived Hourly Pma Status*** == false))

Case (***Current DHV Parameter***)

SO2: ***SO2 Derived Hourly Status*** = false
NOXR: ***NOXR Derived Hourly Status*** = false
NOX: ***NOX Derived Hourly Status*** = false
CO2: ***CO2 Derived Hourly Status*** = false
HI: ***HI Derived Hourly Status*** = false
CO2C: ***CO2C Derived Hourly Status*** = false
H2O: ***H2O Derived Hourly Status*** = false
SO2R: ***SO2R Derived Hourly Status*** = false
SO2M: ***SO2M Derived Hourly Status*** = false
NOXM: ***NOXM Derived Hourly Status*** = false
CO2M: ***CO2M Derived Hourly Status*** = false
HIT: ***HIT Derived Hourly Status*** = false

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2 Mass Rate Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ H2O Derived Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ Heat Input Derived Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ NOx Mass Rate Derived Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ SO2 Derived Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ SO2R Derived Hourly Evaluation

Check Code: HOURDHV-36

Check Name: NOx Rate DHV Extraneous Fields Check

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Hourly Extraneous Fields = null

if (***Current DHV Method*** <> "AE" OR (***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil*** > 0 AND ***App E Constant Fuel Mix*** == false)
    if (***Current DHV Record.***SegmentNumber is not null)
        append "SegmentNumber" to ***Hourly Extraneous Fields***

    if (***Current DHV Record.***OperatingConditionCode is not null)
        append "OperatingConditionCode" to ***Hourly Extraneous Fields***

if (***Current DHV Method*** <> "LME")
    if (***Current DHV Record.***FuelCode is not null)
        append "FuelCode" to ***Hourly Extraneous Fields***

if (***Hourly Extraneous Fields*** is not null),
    return result A

Results:

Result	Response	Severity
A	You reported [fieldnames] in the DHV record for [param]. This data should be blank.	Non-Critical Error

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Emissions Rate Derived Hourly Evaluation

Check Code: HOURDHV-37

Check Name: Calculate Heat Input for LME Unit

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

HIT Calculated Adjusted Value = null

If (***Derived Hourly Modc Status*** == true)
if (***LME HI Method*** is equal to "MHHI" OR ***Current DHV Record***.MODCCode = "45")

Locate all ***Monitor Default*** records for the hour and location where the ParameterCode is equal to "MHHI".

If (one record is found, AND ***Monitor Default.***DefaultValue is greater than 0, AND ***Monitor Default.***DefaultUnitsOfMeasureCode is equal to "MMBTUHR")
If (***Current Hourly Op Record***.OpTime is greater than 0 and less than or equal to 1)
Calculate ***HIT Calculated Adjusted Value*** = DefaultValue * ***Current Hourly Op Record***.OpTime, rounded to one decimal place.

else
return result A

else if (***LME HI Method*** is equal to "LTFF")

If (***LME CP Total Heat Input*** is greater than or equal to 0, AND ***LME Total Heat Input Array*** for the location is greater than or equal to 0, AND ***Current Hourly Op Record***.HourLoad is greater than or equal to 0, AND ***Current Hourly Op Record***.OpTime is greater than 0 and less than or equal to 1)

If (***LME OS*** is equal to true, AND the Quarter of the ***Current Reporting Period*** is equal to 2)

If the ***Current Month*** is April,

If (***LME April Load*** is greater than 0)

If (HourLoad is equal to 0)
Set ***HIT Calculated Adjusted Value*** = 0

else
Calculate ***HIT Calculated Adjusted Value*** = (***LME CP April Heat Input*** * ***Current Hourly Op Record***.HourLoad * ***Current Hourly Op Record***.OpTime / ***LME April Load***) + (***LME April Heat Input Array*** for the location * ***Current Hourly Op Record***.HourLoad * ***Current Hourly Op Record***.OpTime / ***LME April Load Array*** for the location), and round the result to 1 decimal place.

else if (***LME April Optime*** is greater than 0)

Calculate ***HIT Calculated Adjusted Value*** = (***LME CP April Heat Input*** * ***Current Hourly Op Record***.OpTime / ***LME April Optime***) + (***LME April Heat Input Array*** for the location * ***Current Hourly Op Record***.OpTime / ***LME April OpTime Array*** for the location), and round the result to 1 decimal place.

Otherwise,

If (***LME Total Load*** is greater than 0)

If (HourLoad is equal to 0)
Set ***HIT Calculated Adjusted Value*** = 0

else
Calculate ***HIT Calculated Adjusted Value*** = ((***LME CP Total Heat Input - LME CP April Heat Input***) * ***Current Hourly Op Record***.HourLoad * ***Current Hourly Op Record***.OpTime / (***LME Total Load - LME April Load***)) + ((***LME Total Heat Input Array*** for the location - ***LME April Heat Input Array*** for the location) * ***Current Hourly Op Record***.HourLoad * ***Current Hourly Op Record***.OpTime / (***LME Total Load Array*** for the location - ***LME April Load Array*** for the location)), and round the result to 1 decimal place.

else if (***LME Total Optime*** is greater than 0)

Calculate ***HIT Calculated Adjusted Value*** = ((***LME CP Total Heat Input - LME CP April Heat Input***) * ***Current Hourly Op Record***.OpTime / (***LME Total Optime - LME April Optime***)) + ((***LME Total Heat Input Array*** for the location - ***LME April Heat Input Array*** for the location) * ***Current Hourly Op Record***.OpTime / (***LME Total OpTime Array*** for the location - ***LME April OpTime Array*** for the location)), and round the result to 1 decimal place.

Otherwise,

If (***LME Total Load*** is greater than 0)

If (HourLoad is equal to 0)
Set ***HIT Calculated Adjusted Value*** = 0

else
Calculate ***HIT Calculated Adjusted Value*** = (***LME CP Total Heat Input * Current Hourly Op Record***.HourLoad * ***Current Hourly Op Record***.OpTime / ***LME Total Load***) + (***LME Total Heat Input Array*** for the location * ***Current Hourly Op Record***.HourLoad * ***Current Hourly Op Record***.OpTime / ***LME Total Load Array*** for the location), and round the result to 1 decimal place.

else if (***LME Total Optime*** is greater than 0)

Calculate ***HIT Calculated Adjusted Value*** = (***LME CP Total Heat Input * Current Hourly Op Record***.OpTime / ***LME Total Optime***) + (***LME Total Heat Input Array*** for the location * ***Current Hourly Op Record***.OpTime / ***LME Total OpTime Array*** for the location), and round the result to 1 decimal place.

Results:

Result	Response	Severity
A	You did not report a single, active, valid default record for MHHI in your monitoring plan.	Critical Error Level 1
B	This check result is obsolete.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ HIT Derived Hourly Evaluation (LME)

Check Code: HOURDHV-38

Check Name: Check Reported Heat Input for LME Unit

Related Former Checks:

Applicability: LME Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

If (***Current DHV Record***.AdjustedHourlyValue is null or is less than 0)
Rpt Period HI Reported Accumulator Array for this location = -1
return result A
else if (***Current DHV Record***.AdjustedHourlyValue is not rounded to one decimal place)
Rpt Period HI Reported Accumulator Array for this location = -1
return result C
else
if (***Current Month*** is not April OR ***LME Annual*** == true)
if (***Rpt Period HI Reported Accumulator Array*** for this location is not null)
if (***Rpt Period HI Reported Accumulator Array*** >= 0)
Rpt Period HI Reported Accumulator Array for this location = ***Rpt Period HI Reported Accumulator Array*** for this location + ***Current DHV Record***.AdjustedHourlyValue
else
Rpt Period HI Reported Accumulator Array for this location = ***Current DHV Record***.AdjustedHourlyValue

If (***HIT Calculated Adjusted Value*** is not null and ***Current DHV Record***.AdjustedHourlyValue is not equal to ***HIT Calculated Adjusted Value)***

If (***HIT Calculated Adjusted Value*** is greater than 1 OR ***Current DHV Record***.AdjustedHourlyValue is greater than 1)

Heat Input Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "HIT" AND
UOM = "MMBTU"

if (ABS(***Current DHV Record***.AdjustedHourlyValue - ***HIT Calculated Adjusted Value***) > *Heat Input Tolerance*)
return result B.

Results:

Result	Response	Severity
A	The AdjustedHourlyValue reported in the DHV record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
B	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1
C	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ HIT Derived Hourly Evaluation (LME)

Check Code: HOURDHV-39

Check Name: Calculate SO2 Mass for LME Unit

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

SO2M Calculated Adjusted Value = null

If ***Current DHV Record***.FuelCode is null,
Rpt Period SO2 Mass Calculated Accumulator Array for this location = -1
return result A.

Otherwise,
Locate MonitorDefault record for the hour and location where ParameterCd = "SO2R", DefaultPurposeCd = "LM", and FuelCode is equal to ***Current DHV Record***.FuelCode.

If not found, or if more than one record is found, or if DefaultValue is less than or equal to 0, or DefaultValueUnitsOfMeasure is not equal to "LBMMBTU".
Rpt Period SO2 Mass Calculated Accumulator Array for this location = -1
return result B.

Otherwise,
SO2R Default Value = ***MonitorDefault***.DefaultValue

Locate ***MonitorDefault*** record for the hour and location where ParameterCd = "SO2R", DefaultPurposeCd = "LM", FuelCode is in ***LME Fuel Code List***, FuelCode is not equal to ***Current DHV Record***.FuelCode, Default Value is greater than *SO2R Default Value*, and DefaultValueUnitsOfMeasure is equal to "LBMMBTU".

If found,
Rpt Period SO2 Mass Calculated Accumulator Array for this location = -1
return result C.

Otherwise,
If ***HIT Calculated Adjusted Value*** is null,
Rpt Period SO2 Mass Calculated Accumulator Array for this location = -1
return result D.

else
Calculate ***SO2M Calculated Adjusted Value = HIT Calculated Adjusted Value ****** *SO2R Default Value*, and round the result to one decimal place.

if (***Rpt Period SO2 Mass Calculated Accumulator Array*** for this location is not null)
if (***Rpt Period SO2 Mass Calculated Accumulator Array*** for this location >= 0)
Rpt Period SO2 Mass Calculated Accumulator Array for this location = ***Rpt Period SO2 Mass Calculated Accumulator Array*** for this location + ***SO2M Calculated Adjusted Value***
else
Rpt Period SO2 Mass Calculated Accumulator Array for this location = ***SO2 Mass Calculated Adjusted Value***

Results:

Result	Response	Severity
A	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated, because you did not report a FuelCode in this record.	Critical Error Level 1
B	You have not reported one and only one active Monitor Default record with a valid ParameterCode and DefaultPurposeCode in your monitoring plan to report the default emission rate for the fuel. The AdjustedHourlyValue in the DHV for [param] could not be recalculated.	Critical Error Level 1
C	You reported [Fuel] as the FuelCode in the DHV record for [param], but, according to the Monitor Default records in your monitoring plan, this fuel does not have the highest default emissions rate of the fuels combusted during the hour. The AdjustedHourlyValue could not be recalculated.	Critical Error Level 1
D	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated because the heat input rate could not be determined for the hour.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ SO2M Derived Hourly Evaluation (LME)

Check Code: HOURDHV-40

Check Name: Determine Fuels Burned for LME Unit

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

LME Fuel Code List = null

If (***HIT Calculated Adjusted Value*** is not null)

Locate all ***DerivedHourlyValue*** records for the location and hour where ParameterCode in set {SO2M, CO2M, NOXM}

For each record found,
Append ***DerivedHourlyValue***.FuelCode to ***LME Fuel Code List***.

if (***Current Month*** is not April OR ***LME Annual*** == true)
if (***Rpt Period HI Calculated Accumulator Array*** for this location is not null)
if (***Rpt Period HI Calculated Accumulator Array*** for this location >= 0)
Rpt Period HI Calculated Accumulator Array for this location = ***Rpt Period HI Calculated Accumulator Array*** for this location + ***HIT Calculated Adjusted Value***
else
Rpt Period HI Calculated Accumulator Array for this location = ***HIT Calculated Adjusted Value***

if (***Current Month*** is April)
if (***April HI Calculated Accumulator Array*** for this location is not null)
April HI Calculated Accumulator Array for this location = ***April HI Calculated Accumulator Array*** for this location + ***HIT Calculated Adjusted Value***
else
April HI Calculated Accumulator Array for this location = ***HIT Calculated Adjusted Value***

else
if (***Current Month*** is not April OR ***LME Annual*** == true)
Rpt Period HI Calculated Accumulator Array for this location = -1

return result A

Results:

Result	Response	Severity
A	The AdjustedHourlyValue in the DHV record for HIT could not be recalculated due to another error listed in this report.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ HIT Derived Hourly Evaluation (LME)

Check Code: HOURDHV-41

Check Name: Check Reported SO2M for LME Unit

Related Former Checks:

Applicability: LME Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

If (***Current DHV Record***.AdjustedHourlyValue is null or is less than 0)
Rpt Period SO2 Mass Reported Accumulator Array for this location = -1
return result A
else if (***Current DHV Record***.AdjustedHourlyValue is not rounded to one decimal place)
Rpt Period SO2 Reported Accumulator Array for this location = -1
return result C
else
if (***Rpt Period SO2 Mass Reported Accumulator Array*** for this location is not null)
if (***Rpt Period SO2 Mass Reported Accumulator Array*** >= 0)
Rpt Period SO2 Mass Reported Accumulator Array for this location = ***Rpt Period SO2 Mass Reported Accumulator Array*** for this location + ***Current DHV Record***.AdjustedHourlyValue
else
Rpt Period SO2 Mass Reported Accumulator Array for this location = ***Current DHV Record***.AdjustedHourlyValue

If (***SO2M Calculated Adjusted Value*** is not null AND ***Current DHV Record***.AdjustedHourlyValue is not equal to ***SO2M Calculated Adjusted Value)***

SO2 Mass Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "SO2M" AND
UOM = "LB"

if (ABS(***Current DHV Record***.AdjustedHourlyValue - ***SO2M Calculated Adjusted Value***) > *SO2 Mass Tolerance*)
return result B

Results:

Result	Response	Severity
A	The AdjustedHourlyValue reported in the DHV record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
B	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1
C	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ SO2M Derived Hourly Evaluation (LME)

Check Code: HOURDHV-42

Check Name: Calculate CO2 Mass for LME Unit

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

CO2M Calculated Adjusted Value = null

If ***Current DHV Record***.FuelCode is null,
Rpt Period CO2 Mass Calculated Accumulator Array for this location = -1
return result A.

Otherwise,
Locate MonitorDefault record for the hour and location where ParameterCd = "CO2R", DefaultPurposeCd = "LM", and FuelCode is equal to ***Current DHV Record***.FuelCode.

If not found, or if more than one record is found, or if DefaultValue is less than or equal to 0, or DefaultValueUnitsOfMeasure is not equal to "TNMMBTU".
Rpt Period CO2 Mass Calculated Accumulator Array for this location = -1
return result B.

Otherwise,
CO2R Default Value = ***MonitorDefault***.DefaultValue

Locate ***MonitorDefault*** record for the hour and location where ParameterCd = "CO2R", DefaultPurposeCd = "LM", FuelCode is in ***LME Fuel Code List***, FuelCode is not equal to ***Current DHV Record***.FuelCode, Default Value is greater than *CO2R Default Value*, and DefaultValueUnitsOfMeasure is equal to "TNMMBTU".

If found,
Rpt Period CO2 Mass Calculated Accumulator Array for this location = -1
return result C.

Otherwise,
If ***HIT Calculated Adjusted Value*** is null,
Rpt Period CO2 Mass Calculated Accumulator Array for this location = -1
return result D.

else
Calculate ***CO2M Calculated Adjusted Value = HIT Calculated Adjusted Value *** *CO2R Default Value*, and round the result to one decimal place.

if (***Rpt Period CO2 Mass Calculated Accumulator Array*** for this location is not null)
if (***Rpt Period CO2 Mass Calculated Accumulator Array*** for this location >= 0)
Rpt Period CO2 Mass Calculated Accumulator Array for this location = ***Rpt Period CO2 Mass Calculated Accumulator Array*** for this location + ***CO2M Calculated Adjusted Value***
else
Rpt Period CO2 Mass Calculated Accumulator Array for this location = ***CO2 Mass Calculated Adjusted Value***

Results:

Result	Response	Severity
A	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated, because you did not report a FuelCode in this record.	Critical Error Level 1
B	You have not reported one and only one active Monitor Default record with a valid ParameterCode and DefaultPurposeCode in your monitoring plan to report the default emission rate for the fuel. The AdjustedHourlyValue in the DHV for [param] could not be recalculated.	Critical Error Level 1
C	You reported [Fuel] as the FuelCode in the DHV record for [param], but, according to the Monitor Default records in your monitoring plan, this fuel does not have the highest default emissions rate of the fuels combusted during the hour. The AdjustedHourlyValue could not be recalculated.	Critical Error Level 1
D	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated because the heat input rate could not be determined for the hour.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2M Derived Hourly Evaluation (LME)

Check Code: HOURDHV-43

Check Name: Check Reported CO2M for LME Unit

Related Former Checks:

Applicability: LME Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

If (***Current DHV Record***.AdjustedHourlyValue is null or is less than 0)
Rpt Period CO2 Mass Reported Accumulator Array for this location = -1
return result A
else if (***Current DHV Record***.AdjustedHourlyValue is not rounded to one decimal place)
Rpt Period CO2 Reported Accumulator Array for this location = -1
return result C
else
if (***Rpt Period CO2 Mass Reported Accumulator Array*** for this location is not null)
if (***Rpt Period CO2 Mass Reported Accumulator Array*** >= 0)
Rpt Period CO2 Mass Reported Accumulator Array for this location = ***Rpt Period CO2 Mass Reported Accumulator Array*** for this location + ***Current DHV Record***.AdjustedHourlyValue
else
Rpt Period CO2 Mass Reported Accumulator Array for this location = ***Current DHV Record***.AdjustedHourlyValue

If (***CO2M Calculated Adjusted Value*** is not null AND ***Current DHV Record***.AdjustedHourlyValue is not equal to ***CO2M Calculated Adjusted Value)***

CO2 Mass Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "CO2M" AND
UOM = "TON"

if (ABS(***Current DHV Record***.AdjustedHourlyValue - ***CO2M Calculated Adjusted Value***) > *CO2 Mass Tolerance*)
return result B

Results:

Result	Response	Severity
A	The AdjustedHourlyValue reported in the DHV record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
B	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1
C	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ CO2M Derived Hourly Evaluation (LME)

Check Code: HOURDHV-44

Check Name: Calculate NOX Mass for LME Unit

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

NOXM Calculated Adjusted Value = null
UDEFStatus = null.
UDEFExpirationDate = null.

If ***Current DHV Record***.FuelCode is null,
    if (***Current Month*** is not April OR ***LME Annual*** == true)
        Rpt Period NOx Mass Calculated Accumulator Array for this location = -1

    return result A

Otherwise,

    Default Condition = null

    If ***Current DHV Record***.OperatingConditionCode is null,
        Default Condition = "A"

    else if ***Current DHV Record***.OperatingConditionCode in set {C, U, P, B}
        Default Condition = ***Current DHV Record***.OperatingConditionCode

    If *Default Condition* is null,
        if (***Current Month*** is not April OR ***LME Annual*** == true)
            Rpt Period NOx Mass Calculated Accumulator Array for this location = -1

        return result B

    else

        if (***Current DHV Record***.OperatingConditionCode is equal to "U")
            Locate MonitorDefault record for the hour and location where ParameterCd = "NORX", DefaultPurposeCd = "MD", OperatingConditionCode is equal to *Default Condition*, and FuelCode is equal to ***Current DHV Record***.FuelCode.

        else
            Locate MonitorDefault record for the hour and location where ParameterCd = "NOXR", DefaultPurposeCd = "LM", OperatingConditionCode is equal to *Default Condition*, and FuelCode is equal to ***Current DHV Record***.FuelCode.

        If not found, or if more than one record is found, or if DefaultValue is less than or equal to 0, or DefaultValueUnitsOfMeasure is not equal to "LBMMBTU".
            if (***Current Month*** is not April OR ***LME Annual*** == true)
                Rpt Period NOx Mass Calculated Accumulator Array for this location = -1

            return result C

        Otherwise,
            NOXR Default Value = ***MonitorDefault***.DefaultValue

            if (*Default Condition* is in set {A,C,B,P} AND ***MonitorDefault***.DefaultSourceCode == "TEST")

if (***MonitorDefault***.GroupID is null)

if (*Default Condition* == "A" or "C")

Locate the latest ***UnitDefaultTestRecordsByLocationForQAStatus*** for the location where FuelCode = ***Current DHV Record***.FuelCode and EndDate/EndHour is on or before the ***CurrentOperatingDate/Hour***.

else if (*Default Condition* == "B")

Locate the latest ***UnitDefaultTestRecordsByLocationForQAStatus*** for the location where FuelCode = ***Current DHV Record***.FuelCode, OperatingConditionCode == "A" or "B", and EndDate/EndHour is on or before the ***CurrentOperatingDate/Hour***.

else if (*Default Condition* == "P")

Locate the latest ***UnitDefaultTestRecordsByLocationForQAStatus*** for the location where FuelCode = ***Current DHV Record***.FuelCode, OperatingConditionCode == "A" or "P", and EndDate/EndHour is on or before the ***CurrentOperatingDate/Hour***.

If not found,
UDEFStatus = "MISSING"

else
UDEFStatus = "FOUND"
Set ***UDEFExpiration Date*** to 5 years after the end of the quarter of the ***UnitDefaultTestRecordsByLocationForQAStatus***.EndDate.

Otherwise,

UDEFStatus = "GROUP"
Set ***UDEFExpiration Date*** to 5 years after the end of the quarter of the ***MonitorDefault***.BeginDate.

if (***Current DHV Record***.OperatingConditionCode is equal to "U")
Locate ***MonitorDefault*** record for the hour and location where ParameterCd = "NORX", DefaultPurposeCd = "MD", OperatingConditionCode is equal to *Default Condition*, FuelCode is in ***LME Fuel Code List***, FuelCode is not equal to ***Current DHV Record***.FuelCode, Default Value is greater than *NOXR Default Value*, and DefaultValueUnitsOfMeasure is equal to "LBMMBTU".

else
Locate ***MonitorDefault*** record for the hour and location where ParameterCd = "NOXR", DefaultPurposeCd = "LM", OperatingConditionCode is equal to *Default Condition*, FuelCode is in ***LME Fuel Code List***, FuelCode is not equal to ***Current DHV Record***.FuelCode, Default Value is greater than *NOXR Default Value*, and DefaultValueUnitsOfMeasure is equal to "LBMMBTU".

If found,
if (***Current Month*** is not April OR ***LME Annual*** == true)
Rpt Period NOx Mass Calculated Accumulator Array for this location = -1

return result D

Otherwise,

If ***HIT Calculated Adjusted Value*** is null,
if (***Current Month*** is not April OR ***LME Annual*** == true)

Rpt Period NOx Mass Calculated Accumulator Array for this location = -1

return result E

else

Calculate ***NOXM Calculated Adjusted Value = HIT Calculated Adjusted Value *** *NOXR Default Value*, and round the result to one decimal place.

if (***Current Month*** is not April OR ***LME Annual*** == true)

if (***Rpt Period NOx Mass Calculated Accumulator Array*** for this location is not null)
if (***Rpt Period NOx Mass Calculated Accumulator Array*** for this location >= 0)
Rpt Period NOx Mass Calculated Accumulator Array for this location = ***Rpt Period NOx Mass Calculated Accumulator Array*** for this location + ***NOXM Calculated Adjusted Value***
else
Rpt Period NOx Mass Calculated Accumulator Array for this location = ***NOx Mass Calculated Adjusted Value***

if (***Current Month*** is April)
if (***April NOx Mass Calculated Accumulator Array*** for this location is not null)
April NOx Mass Calculated Accumulator Array for this location = ***April NOx Mass Calculated Accumulator Array*** for this location + ***NOXM Calculated Adjusted Value***
else
April NOx Mass Calculated Accumulator Array for this location = ***NOXM Calculated Adjusted Value***

Results:

Result	Response	Severity
A	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated, because you did not report a FuelCode in this record.	Critical Error Level 1
B	You reported an invalid OperatingConditionCode in the DHV record for [param]. The AdjustedHourlyValue could not be recalculated.	Critical Error Level 1
C	You have not reported one and only one active Monitor Default record with a valid ParameterCode, DefaultPurposeCode, and OperatingConditionCode in your monitoring plan to report the default emission rate for the fuel. The AdjustedHourlyValue in the DHV for [param] could not be recalculated.	Critical Error Level 1
D	You reported [Fuel] as the FuelCode in the DHV record for [param], but, according to the Monitor Default records in your monitoring plan, this fuel does not have the highest default emissions rate of the fuels combusted during the hour. The AdjustedHourlyValue could not be recalculated.	Critical Error Level 1
E	The AdjustedHourlyValue in the DHV record for [param] could not be recalculated because the heat input rate could not be determined for the hour.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ NOXM Derived Hourly Evaluation (LME)

Check Code: HOURDHV-45

Check Name: Check Reported NOXM for LME Unit

Related Former Checks:

Applicability: LME Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

If (***Current DHV Record***.AdjustedHourlyValue is null or is less than 0)
Rpt Period NOx Mass Reported Accumulator Array for this location = -1
return result A
else if (***Current DHV Record***.AdjustedHourlyValue is not rounded to one decimal place)
Rpt Period NOx Mass Reported Accumulator Array for this location = -1
return result C
else
if (***Current Month*** is not April OR ***LME Annual*** == true)
if (***Rpt Period NOx Mass Reported Accumulator Array*** for this location is not null)
if (***Rpt Period NOx Mass Reported Accumulator Array*** >= 0)
Rpt Period NOx Mass Reported Accumulator Array for this location = ***Rpt Period NOx Mass Reported Accumulator Array*** for this location + ***Current DHV Record***.AdjustedHourlyValue
else
Rpt Period NOx Mass Reported Accumulator Array for this location = ***Current DHV Record***.AdjustedHourlyValue

If (***NOXM Calculated Adjusted Value*** is not null AND ***Current DHV Record***.AdjustedHourlyValue is not equal to ***NOXM Calculated Adjusted Value)***

NOX Mass Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "NOXM" AND
UOM = "LB"

if (ABS(***Current DHV Record***.AdjustedHourlyValue - ***NOXM Calculated Adjusted Value***) > *NOX Mass Tolerance*)
return result B

Results:

Result	Response	Severity
A	The AdjustedHourlyValue reported in the DHV record for [param] is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
B	The AdjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1
C	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ NOXM Derived Hourly Evaluation (LME)

Check Code: HOURDHV-46

Check Name: Equation Code Consistent with Moisture Basis

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ SO2M Derived Hourly Evaluation (LME)

Check Code: HOURDHV-47

Check Name: Unit Default Test Expiration Check

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

If ***UDEF Status*** is not null,

If (***UDEF Status*** == "MISSING")
return result A.

else if (***UDEF Status*** == "FOUND")
if (***Current Operating Date*** is after the ***UDEF Expiration Date***)
return result B.

else
Append ***CurrentDHV***.FuelCode to the ***LME Fuel Array*** for the location.

else if (***UDEF Status*** == "GROUP")
if (***Current Operating Date*** is after the ***UDEF Expiration Date***)
return result C.

Results:

Result	Response	Severity
A	You did not report an applicable prior LME Unit Default Test for Fuel Code [FUEL].	Critical Error Level 1
B	The applicable prior LME Unit Default Test for Fuel Code [FUEL] has expired. You need to use a Part 75 default NOx emissions rate until you perform a new unit-and-fuel-specific default test. You will need to put an end date on your existing NOXR default records in your monitoring plan, and add a new NOXR default record based on the Part 75 default value.	Critical Error Level 1
C	Warning: Based on the BeginDate in your NOXR Default record in your monitoring plan, the LME Unit Default Test(s) that established the default NOx emission rate for Fuel Code [FUEL] may have expired. Unit Default Tests must be performed every five years. If your test has expired, you need to use a Part 75 default NOx emissions rate until you perform a new unit-and-fuel-specific default test. You will need to put an end date on your existing NOXR default records in your monitoring plan, and add a new NOXR default record based on the Part 75 default value.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ NOXM Derived Hourly Evaluation (LME)

Check Category:

Hourly General

Check Code: HOURGEN-1

Check Name: Initialize Accumulators, Dictionaries and Lists

Related Former Checks: HOUROP-27

Applicability: General Check

Description: Initializes summary value data: the operating time, operating hours, and reported and calculated values for each parameter. Also initializes dictionaries and lists used to collect information during processing.

Specifications:

For each location in Monitoring Plan, initialize arrays with size ***Current Location Count***

Rpt Period CO2 Mass Reported Accumulator Array for the location = 0
Rpt Period CO2 Mass Calculated Accumulator Array for the location = 0
Expected Summary Value CO2 Array for the location = false

Rpt Period HI Reported Accumulator Array for the location = 0
Rpt Period HI Calculated Accumulator Array for the location = 0
Expected Summary Value HI Array for the location = false

Rpt Period NOx Rate Reported Accumulator Array for the location = 0
Rpt Period NOx Rate Calculated Accumulator Array for the location = 0
Rpt Period NOx Rate Hours Accumulator Array for the location = 0
Expected Summary Value NOx Array for the location = false

Rpt Period SO2 Mass Reported Accumulator Array for the location = 0
Rpt Period SO2 Mass Calculated Accumulator Array for the location = 0
Expected Summary Value SO2 Array for the location = false

Rpt Period NOx Mass Reported Accumulator Array for the location = 0
Rpt Period NOx Mass Calculated Accumulator Array for the location = 0
Expected Summary Value NOx Mass Array for the location = false

Rpt Period Op Time Accumulator Array for the location = 0
Rpt Period Op Hours Accumulator Array for the location = 0
Rpt Period Op Days Accumulator Array for the location = 0
Rpt Period Load Accumulator Array for the location = 0

Daily Op Time Accumulator Array for this location = 0

April HI Calculated Accumulator Array for the location = 0
April NOx Mass Calculated Accumulator Array for the location = 0
April Op Time Accumulator Array for the location = 0
April Op Hours Accumulator Array for the location = 0
April Op Days Accumulator Array for the location = 0

LME Total Load Array for the location = 0
LME April Load Array for the location = 0
LME Total Heat Input Array for the location = 0
LME April Heat Input Array for the location = 0
LME Total OpTime Array for the location = 0
LME April OpTime Array for the location = 0

Last Day of Operation Array for the location = null
First Day of Operation = null
First Hour of Operation = null

FLOW System ID Array for the location = null
NOXE System ID Array for the location = null

LME Fuel Array for the location = null

Operating Date Array for the location = empty date list

Count the number of unique location + FuelCode in the ***Hourly Fuel Flow*** records for the monitoring configuration and reporting period.
Initialize an array with this number of elements:
Fuel Op Hours Accumulator Array for the location and FuelCode = 0

Initialize ***F2LStatusSystemResultDictionary*** as a dictionary with both a string key and lookup value
Initialize ***F2LStatusSystemCheckDictionary*** as a dictionary with a string key and a data row value
Initialize ***F2LStatusSystemMissingOpDictionary*** as a dictionary with both a string key and lookup value

Initialize ***InvalidCylinderIdList*** as a list with string values.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-2

Check Name: Reporting Period Details

Related Former Checks:

Applicability: General Check

Description: Checks the current reporting period to see if the monitoring plan is active. Also sets a parameter indicating whether legacy data is being processed.

Specifications:

Abort Hourly Checks = false
Legacy Data Evaluation = false
LME HI Method = null
LME HI Substitute Data Code = null
Annual Reporting Requirement = false
OS Reporting Requirement = false
LME Annual = false
LME OS = false
Reported Emissions Value = null
Multiple Stack Configuration = false
Ignored Daily Calibration Tests = false
Ignored Daily Interference Tests = false

if ((***Current Reporting Period*** < ***Current Monitoring Plan Record***.BeginReportPeriod) OR
(***Current Monitoring Plan Record***.EndReportPeriod is not null AND ***Current Monitoring Plan Record***.EndReportPeriod < ***Current Reporting Period***))

Abort Hourly Checks = true
return result A
else

Locate a ***UnitProgram*** record for any unit in the configuration where ProgramCode in ***Program is Ozone Season List***, UnitMonitorCertBeginDate is on or before the December 31 of the year of the ***Current Reporting Period***, and the EndDate is null or is on or after Jan 1 of the year of the ***Current Reporting Period.***

If found,
OS Reporting Requirement = true

Locate all ***LocationReportingFrequency*** record for any unit in the configuration where BeginQuarter is on or before the ***Current Reporting Period***, and the EndQuarter is null or is on or after the ***Current Reporting Period***.

If found, and the ReportingFrequencyCode in all records == "Q",
Annual Reporting Requirement = true

else if found, and the ReportingFrequencyCode in all records == "OS",
If ***OS Reporting Requirement*** == false
Abort Hourly Checks = true
return result B

else if the Quarter of the ***Current Reporting Period*** is equal to 1 or 4,
Abort Hourly Checks = true
return result C

Otherwise,
Abort Hourly Checks = true
return result B

If (***Abort Hourly Checks*** == false)

If (the associated First ECMPS Reporting Period for the monitoring plan is null)
If ***Current Reporting Period*** is on or prior to 2008)
Legacy Data Evaluation = true
Else
If ***Current Reporting Period*** is prior to the First ECMPS Reporting Period)
Legacy Data Evaluation = true

Locate a Hourly Op Data record for the configuration and reporting period where OperatingTime is greater than 0,

If found,
Reporting Period Operating = true

else,
Reporting Period Operating = false

Set ***LME Annual*** to false.
Set ***LME OS*** to false.

Set *AnyMonitoringMethodFound* to false.
Set *OsMonitoringMethodFound* to false.

Locate MonitorMethod record for ANY location in the file where:

1) ParameterCode in set {SO2M, NOXM, CO2M}.
2) MethodCode = "LME".
3) BeginDate is on or before the first day of the ***Current Reporting Period***.
4) EndDate is null or is on or after the last day of the ***Current Reporting Period***.

If found,

Set *AnyMonitoringMethodFound* to true

If ***Current Reporting Period*** is for the 2nd or 3rd Quarter,
Set *OsMonitoringMethodFound* to true

Locate all MonitorQualification records for all units in the monitoring configuration where:

1) QualificationTypeCode is equal to "LMEA".
2) BeginDate is on or before the last day of the reporting period.
3) EndDate is null or is on or after January 1 of the year of the ***Current Reporting Period***.

If found,
Set ***LME Annual*** to true.

If *OsMonitoringMethodFound* is false, AND ***Current Reporting Period*** is for the 2nd Quarter,

Locate MonitorMethod record for ANY location in the file where:

1) ParameterCode in set {SO2M, NOXM, CO2M}.
2) MethodCode = "LME".
3) BeginDate is on or before May 1st of the year of the ***Current Reporting Period***.
4) EndDate is null OR is on or after the last day of the ***Current Reporting Period***.

If found,
Set *AnyMonitoringMethodFound* to true.
Set *OsMonitoringMethodFound* to true.

If *OsMonitoringMethodFound* ,

Locate all MonitorQualification records for all units in the monitoring configuration where:

1) QualificationTypeCode is equal to "LMES".
2) BeginDate is on or before the last day of the reporting period.
3) EndDate is null OR is on or after January 1 of the year of the ***Current Reporting Period***.

If found,
Set ***LME OS*** to true.

If *AnyMonitoringMethodFound* ,

If (***LME Annual*** == true and ***Annual Reporting Requirement*** == false)
Abort Hourly Checks = true
return result D

else if (***LME OS*** == true and ***OS Reporting Requirement*** == false)
Abort Hourly Checks = true
return result E

else if (***LME Annual*** == false and ***LME OS*** == false)
Abort Hourly Checks = true
return result F

Otherwise,
Locate MonitorMethod records for all locations in the file where:

1) ParameterCode = "HIT".
2) BeginDate is on or before:
a) If ***Current Reporting Period*** is for the 2nd quarter AND ***LME Annual*** is false, then May 1st of the year of the ***Current Reporting Period***.
b) Otherwise, the first day of the ***Current Reporting Period***.
3) EndDate is null OR is on or after the last day of the ***Current Reporting Period***.

If not found for any location,
Abort Hourly Checks = true
return result G

Else

If MethodCode = "MHHI" for all locations,
LME HI Method = "MHHI"

If MethodCode in set {LTFF, CALC, LTFCALC} for all locations,
LME HI Method = "LTFF"

If SubstituteDataCode is equal to "MHHI" for any location,
LME HI Substitute Data Code = "MHHI".

Otherwise,
Abort Hourly Checks = true
return result H

If (***Abort Hourly Checks*** == false)

Locate all Unit Program records for all units in the configuration where the UnitMonitorCertBeginDate is on or prior to the ***Current Reporting Period*** and the EndDate is null or is on or after the ***Current Reporting Period***.

If the ProgramCode in all the retrieved Location Program records is NOT in ***Program Uses RUE List***,

Locate all Unit Operating Status records for all units in the configuration where the Op Status Code is equal to "RET", the year of the Begin Date is prior to ***Current Reporting Period***, and the End Date is null or is on or after the last day of the ***Current Reporting Period***.

If found,
return result I
Otherwise,

For each Unit Program record retrieved above where the ProgramCode is in ***Program Uses RUE List***,

If ProgramCode is in ***Program is Ozone Season List*** and the ***Current Reporting Period*** is in the first or second quarter,

Locate a Unit Program Exemption record for the unit program where the Exempt Type is equal to "RUE", the Exemption Begin Date is on or prior to May 1 of the year of the ***Current Reporting Period***, and the Exemption End Date is null or is on or after the last day of the ***Current Reporting Period***.

Otherwise,

Locate a Unit Program Exemption record for the unit program where the Exempt Type is equal to "RUE", the Exemption Begin Date is on or prior to the first day of the ***Current Reporting Period***, and the Exemption End Date is null or is on or after the last day of the ***Current Reporting Period***.

If a Unit Program Exemption record was found for all unit programs,
return result I

Results:

Result	Response	Severity
A	The locations in the file do not represent a valid monitoring configuration during the reporting period. The file will not be evaluated.	Fatal
B	The active Reporting Frequency records for this configuration are missing or invalid. The file will not be evaluated. Please contact ECMPS technical support for assistance with this matter.	Fatal
C	According to the Reporting Frequency records, this monitoring configuration is an ozone-season-only reporter, however the reporting period is not within the ozone season. The file will not be evaluated.	Fatal
D	You have reported an LME Annual Qualification record in your monitoring plan, but, according to the Reporting Frequency records, this configuration is not an annual reporter. The file will not be evaluated.	Fatal
E	You have reported an LME Ozone Season Qualification record in your monitoring plan, but, according to the Unit Program records, this configuration does not report ozone season totals. The file will not be evaluated.	Fatal
F	You have reported an LME method in your monitoring plan for one or more units in this configuration, but you have not reported an LME qualification record. The file will not be evaluated.	Fatal
G	You did not report an active heat input method for one or more locations in the file. The file will not be evaluated.	Fatal
H	The active heat input methods reported for the locations in the file are inconsistent. The file will not be evaluated.	Fatal
I	This file contains at least one unit that is retired. Please contact EPA if you believe that all units in this configuration should report emissions data during this reporting period.	Critical Error Level 2

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-3

Check Name: Calculate Total Load for LME Configuration for Reporting Period

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

LME Total Load = 0
LME April Load = 0
LME CP Total Heat Input = 0
LME CP April Heat Input = 0
LME Total Optime = 0
LME April Optime = 0

If (***LME HI Method*** is not null)

If (***LME HI Method*** == "MHHI")

Locate an LTFF record for any location in the monitoring configuration during the reporting period.

If found,
set ***Abort Hourly Checks*** to true, and return result A.

else
For each ***Hourly Op Data*** record for every unit in the monitoring configuration.

If ***Hourly Op Data.***OpTime is not equal to 0 or ***Hourly Op Data.***HourLoad is not null,

If ***Hourly Op Data.***HourLoad is greater than or equal to 0 and ***Hourly Op Data.***OpTime is between 0 and 1 (inclusive),

Locate the ***DerivedHourlyValue*** record for the unit and the hour where ParameterCode is equal to "HIT".

If found, AND ***DerivedHourlyValue.***MODCCode is null,
Add HourLoad * OpTime to ***LME Total Load Array*** for the location
Add HourLoad * OpTime to ***LME Total Load***.
Add OpTime to ***LME Total OpTime Array*** for the location
Add OpTime to ***LME Total Optime***.

If the month of ***Hourly Op Data***.Date is "April" AND ***LME OS*** is equal to true ,
Add HourLoad * OpTime to ***LME April Load Array*** for the location
Add HourLoad * OpTime to ***LME April Load***.
Add OpTime to ***LME April OpTime Array*** for the location
Add OpTime to ***LME April Optime***.

If not found, AND ***Hourly Op Data.***HourLoad is greater than 0,
set ***LME Total Load*** to -1.
exit for.

Otherwise,
set ***LME Total Load*** to -1.
exit for.

If ***LME OS*** is equal to true and the reporting period is the second quarter,

Locate an LTFF record for any location in the monitoring configuration during the reporting period where the FuelFlowPeriodCode is equal to "A".

If found, AND ***LME April Load*** is equal to 0 and ***LME April Optime*** is equal to 0,
set ***Abort Hourly Checks*** to true, and return result C.

else if not found AND (***LME April Load*** is greater than 0 or ***LME April Optime*** is greater than 1),
return result F.

else
Locate an LTFF record for any location in the monitoring configuration during the reporting period where the FuelFlowPeriodCode is equal to "MJ".

If found,
If (***LME Total Load - LME April Load***) is equal to 0 and (***LME Total Optime - LME April Optime***) is equal to 0,
set ***Abort Hourly Checks*** to true, and return result E.

else
If (***LME Total Load - LME April Load***) is greater than 0 or (***LME Total Optime - LME April Optime***) is greater than 1,
return result G.

else

Locate an LTFF record for any location in the monitoring configuration during the reporting period.

If found,
If ***LME Total Load*** is equal to 0 and ***LME Total Optime*** is equal to 0,
set ***Abort Hourly Checks*** to true, and return result B.

else
If ***LME Total Load*** is greater than 0 or ***LME Total Optime*** is greater than 1,
return result D.

Results:

Result	Response	Severity
A	You have reported MHHI as the heat input method for this configuration, but you have reported a long-term fuel flow record.	Fatal
B	You have reported a long-term fuel flow record for this reporting period, but the sum of the load and operating time values in the hourly records (where MHHIIndicator is not equal to 1) are equal to 0.	Fatal
C	You have reported a long-term fuel flow record for April, but the sum of the load and operating time values in the hourly records (where MHHIIndicator is not equal to 1) are equal to 0.	Fatal
D	You have reported LTFF as the heat input method for this configuration, but you have not reported a long-term fuel flow record for this reporting period.	Critical Error Level 1
E	You have reported a long-term fuel flow record for May and June, but the sum of the load and operating time values in the hourly records (where MHHIIndicator is not equal to 1) are equal to 0.	Fatal
F	You have reported LTFF as the heat input method for this ozone-season reporting configuration, but you have not reported a long-term fuel flow record for April.	Critical Error Level 1
G	You have reported LTFF as the heat input method for this ozone-season reporting configuration, but you have not reported a long-term fuel flow record for May/June.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-7

Check Name: Validate LME Eligibility

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

If (***LME HI Method*** is not null)

Set ***LME Exceeding Parameter*** to null.
Set *Final LME Year* to false.

For each unit in the monitoring configuration:

If (***LME Annual*** == true)

Locate the latest ***MonitorQualification*** record where location is the unit being evaluated, the QualificationTypeCode is equal to "LMEA", BeginDate is on or before the first day of the ***Current Reporting Period***, and the EndDate is null or is on or after December 31 of the year prior to the ***Current Reporting Period***.

If (not found)
return result A.

If (found and the BeginDate of the retrieved qualification record is prior to the ***Current Reporting Period Year***)

For each year from three years prior to the ***Current Reporting Period Year*** until the year prior to the ***Current Reporting Year***:

Set *Annual NOx* to 0.
Set *Annual SO2* to 0.

For quarter 1 until quarter 4:

Locate an ***Op Supp Data*** record for the location and quarter/year being checked where the OpTypeCode = "NOXM".

If (found)
add OpValue to *Annual NOx*.

Locate an ***Op Supp Data*** record for the location and quarter/year being checked where the OpTypeCode = "SO2M".

If (found)
add OpValue to *Annual SO2*.

If (the year being evaluated is the year prior to the ***Current Reporting Period Year***)

If (*Annual NOx* is greater than 100 or *Annual SO2* is greater than 25)
set *Final LME Year* to true.

Else

If (*Annual NOx* is greater than 100)
append "Annual NOx" to ***LME Exceeding Parameter***.

If (*Annual SO2* is greater than 25)

append "Annual SO2" to ***LME Exceeding Parameter***.

If (***LME OS*** == true)

Locate the latest ***MonitorQualification*** record where the location is the unit being evaluated, QualificationTypeCode is equal to "LMES", BeginDate is on or before the later of the first day of the ***Current Reporting Period*** and May 1 of the year of the ***Current Reporting Period***, and the EndDate is null or is on or after December 31 of the year prior to the ***Current Reporting Period***.

If (not found)
return result B.

If (found and the BeginDate of the retrieved qualification record is prior to the ***Current Reporting Period Year***)

For each year from three years prior to the ***Current Reporting Period Year*** until the year prior to the ***Current Reporting Period Year***:

Set *OS NOx* to 0.

Locate an ***Op Supp Data*** record for the location and quarter 2 of the year being checked where the OpTypeCode = "NOXMOS".

If found,
add OpValue to *OS NOx*.

Locate an ***Op Supp Data*** record for the location and quarter 3 of the year being checked where the OpTypeCode = "NOXM".

If found,
add OpValue to *OS NOx*.

If (the year being evaluated is the year prior to the ***Current Reporting Period Year***)

If (*OS NOx* is greater than 50)
set *Final LME Year* to true.

Else
If (*OS NOx* is greater than 50)
append "Ozone Season NOx" to ***LME Exceeding Parameter***.

if (***LME Exceeding Parameter*** is not null)
return result C.

else if (*Final LME Year* == true)
return result D.

Results:

Result	Response	Severity
A	You have reported an active LMEA qualification record for this configuration in your monitoring plan, but you have not reported an active LMEA qualification record for at least one unit in the configuration.	Fatal
B	You have reported an active LMES qualification record for this configuration in your monitoring plan, but you have not reported an active LMES qualification record for at least one unit in the configuration.	Fatal
C	You have reported that this configuration has an active LME qualification, but this configuration is no longer eligible to qualify for an LME methodology, because at least one unit in the configuration has exceeded the eligibility limit for [param] in a prior year.	Critical Error Level 2
D	The emissions from at least one unit in this configuration exceeded the applicable number of tons necessary to qualify as an LME unit in the previous reporting year. According to Part 75.19(b), you must install the appropriate monitoring systems by December 31 of this reporting year.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-8

Check Name: Monitor Plan Evaluation Check

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Set ***MpSuccessfullyEvaluated*** = false.

If the SeverityLevelCd of the monitoring plan record for this configuration is equal to "CRIT1" or "FATAL".
return result A.

Otherwise,

If the NeedsEvalFlag and MustSubmitFlag of the monitoring plan record for this configuration are equal to "Y",
return result B.

Otherwise,
Set ***MpSuccessfullyEvaluated*** = true.

Results:

Result	Response	Severity
A	The Monitoring Plan associated with this quarterly emissions file has critical errors. You must correct the Monitoring Plan for this monitoring configuration in order to submit this emissions file to be loaded on EPA's host system.	Critical Error Level 1
B	The Monitoring Plan associated with this quarterly emissions file has not been evaluated. You must evaluate the Monitoring Plan for this monitoring configuration in order to complete the evaluation of this emissions file.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-9

Check Name: QA/Cert Test Evaluation Check

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Locate any QA/Cert Test record for the monitoring configuration where the EndDate is on or before the last day of the ***Current Reporting Period***, MustSubmitFlag is equal to 'Y' or CanSubmitFlag and UpdatedStatusFlag are equal to "Y", and the SeverityCd is equal to "CRIT1" or "FATAL".

If found,
return result A.

Otherwise,

Locate any QA/Cert Test record for the monitoring configuration where the EndDate is on or before the last day of the ***Current Reporting Period***, MustSubmitFlag is equal to 'Y' or CanSubmitFlag and UpdatedStatusFlag are equal to "Y", and the NeedsEvalFlag are equal to "Y".

If found,
return result B.

Otherwise,
Locate any QA Supp Data record for the monitoring configuration without any associated Test Summary record where the EndDate is on or before the last day of the ***Current Reporting Period*** and MustSubmitFlag is equal to 'Y'.

If found,
return result C.

Results:

Result	Response	Severity
A	At least one QA/certification test associated with this monitoring configuration has critical errors. You must correct all prior or concurrent QA/certification tests in order to submit this quarterly emissions file to be loaded on EPA's host system.	Critical Error Level 1
B	At least one QA/certification test associated with this monitoring configuration has not been evaluated. You must evaluate all prior or concurrent QA/certification tests in order to complete the evaluation of this quarterly emissions file.	Critical Error Level 1
C	The emissions quarterly reported cannot be submitted, because EPA has required the resubmission of a QA/certification test that is not present in the Client Tool. Please review the Submission Access report for more information about what needs to be submitted.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-10

Check Name: QA/Cert Event Evaluation Check

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Locate any QA/Certification Event record for the monitoring configuration where the QACertEventDate is on or before the last day of the ***Current Reporting Period***, the MustSubmitFlag is equal to "Y", and the SeverityCd is equal to "CRIT1" or "FATAL".

If found,
return result A.

Otherwise,

Locate any QA/Certification Event record for the monitoring configuration where the QACertEventDate is on or before the last day of the ***Current Reporting Period*** and the MustSubmitFlag and NeedsEvalFlag are equal to "Y".

If found,
return result B.

Results:

Result	Response	Severity
A	At least one QA/certification event associated with this monitoring configuration has critical errors. You must correct all prior or concurrent QA/certification events in order to submit this quarterly emissions file to be loaded on EPA's host system.	Critical Error Level 1
B	At least one QA/certification event associated with this monitoring configuration has not been evaluated. You must evaluate all prior or concurrent QA/certification event in order to complete the evaluation of this quarterly emissions file.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-11

Check Name: Test Extension/Exemption Evaluation Check

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Locate any Test Extension/Exemption record for the monitoring configuration where the ReportingPeriod is on or before the ***Current Reporting Period***, the MustSubmitFlag is equal to "Y", and the SeverityCd is equal to "CRIT1" or "FATAL".

If found,
return result A.

Otherwise,
Locate any Test Extension/Exemption record for the monitoring configuration where the ReportingPeriod is on or before the ***Current Reporting Period*** and the MustSubmitFlag and NeedsEvalFlag are equal to "Y".

If found,
return result B.

Results:

Result	Response	Severity
A	At least one test extension/exemption associated with this monitoring configuration has critical errors. You must correct all prior or concurrent test extension/exemption records in order to submit this quarterly emissions file to be loaded on EPA's host system.	Critical Error Level 1
B	At least one test extension/exemption associated with this monitoring configuration has not been evaluated. You must evaluate all prior or concurrent test extension/exemption records in order to complete the evaluation of this quarterly emissions file.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-12

Check Name: Prior Emissions File Evaluation Check

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Locate any Emissions File for any location in the current monitoring configuration where the ReportingPeriod is prior to the ***Current Reporting Period***, and either the Submission Availability Code = 'CRITERR' or (CAN_SUBMIT = 'Y' and SeverityCd is equal to "CRIT1" or "FATAL").

If found,
return result A.

Otherwise,
Locate any Emissions File for any location in the current monitoring configuration where CAN_SUBMIT = 'Y', the ReportingPeriod is prior to the ***Current Reporting Period***, and the NeedsEvalFlag is equal to "Y".

If found,
return result B.

If not found,
Locate any Emissions File for any location in the current monitoring configuration where the ReportingPeriod is prior to the ***Current Reporting Period***, and either the Submission Availability Code = 'NOTSUB' or (CAN_SUBMIT = 'Y' and UpdatedStatusFlag = 'NODATA').

If found,
return result C.

Results:

Result	Response	Severity
A	At least one prior quarterly emissions file for at least one location in this monitoring configuration has critical errors. You must correct all prior quarterly emissions files in order to submit this quarterly emissions file to be loaded on EPA's host system.	Critical Error Level 1
B	At least one prior quarterly emissions file for at least one location in this monitoring configuration has not been evaluated. You must evaluate all prior quarterly emissions files in order to complete the evaluation for this quarterly emissions file.	Critical Error Level 1
C	At least one prior quarterly emissions file for at least one location in this monitoring configuration has not been submitted and has been authorized for resubmission. You must submit all prior quarterly emissions files in order to submit this quarterly emissions file to be loaded on EPA's host system.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-13

Check Name: Determine If File Can Be Submitted

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Locate the Emission Submission Access record for the configuration and reporting period.

If not found, or the Submission Availability Code is null,
return result A.

else if the Submission Availability Code is not equal to "GRANTED" or "REQUIRE",
return result B.

Results:

Result	Response	Severity
A	The emissions quarterly report cannot be submitted, either because the EPA has not yet opened the submission window, you have not logged into the EPA host system, or you are no longer a representative or agent for this facility. If you are a representative or agent for this facility, when EPA opens the submission window you should log in to the EPA host system to receive automatic permission to submit. You will then need to reevaluate this file prior to submitting.	Informational Message
B	The emissions quarterly report cannot be resubmitted until you contact the EPA for permission. After the EPA grants permission, you will need to log in to the EPA host system to retrieve the permission record. You will then need to reevaluate this file prior to submitting.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-14

Check Name: Ignored Offline Daily Calibration Check

Related Former Checks:

Applicability:

Description:

Specifications:

If (***Ignored Daily Calibration Tests*** == true)
set ***Ignored Daily Calibration Tests*** to false
return result A.

Results:

Result	Response	Severity
A	You reported one or more daily calibration tests that will not fulfill your daily calibration testing requirement, because these tests were performed while the unit was not operating and you have not reported a prior online-offline calibration demonstration. These tests have been assigned a CalculatedTestResult of "IGNORED", and they can be viewed on the Daily Calibration tab of the View Detailed Emissions Screen. If you intend to use offline tests to fulfill your daily calibration testing requirement, you must conduct an online-offline calibration demonstration.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURGEN-15

Check Name: Expiring Test Check

Related Former Checks:

Applicability: General Check

Description:

Specifications:

Set ***Expired Systems*** and ***Expiring Systems*** to null.

Set ***Expiration Text*** to "have expired"

If ***FLOW System ID Array*** for the location is not null,

For each SystemID in the ***FLOW System ID Array*** for the location:

Locate the latest ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the SystemID being checked and the number of operating levels the OpLevelCodeList is equal to 3,

If found,

If ***RATATestRecordsByLocationForQAStatus***.TestReasonCode equal to "INITIAL" then

Locate the latest ***QACertEventsForEMEvaluation*** where SystemID is equal to the SystemID being checked and QACertEventCode equal to "305"

If ***QACertEventforEMEvaluation***.TestCompletionDate is after the ***RATATestRecordsByLocationForQAStatus***.EndDate

Set *ExpirationDate* to five years after the end of the quarter of the ***QACertEventforEMEvaluation***.TestCompletionDate

Else

Set *ExpirationDate* to five years after the end of the quarter of the ***RATATestRecordsByLocationForQAStatus***.EndDate

Else

Set *ExpirationDate* to five years after the end of the quarter of the ***RATATestRecordsByLocationForQAStatus***.EndDate.

If *ExpirationDate* is prior to the current calendar date,

Append the SystemIdentifier being checked to ***Expired Systems***.

else if *ExpirationDate* is on or before the last day of the ***Current Reporting Period***,

Set Expiration Text to "will be expiring at the end of the reporting period".

Append the SystemIdenitifier being checked to ***Expired Systems***.

else if *ExpirationDate* is on or before the last day of the quarter following the ***Current Reporting Period***,

Append the SystemIdentifier being checked to ***Expiring Systems***.

If both ***Expired Systems*** and ***Expiring Systems*** are not null,

return result A

else if ***Expired Systems*** is not null,

return result B

else if ***Expiring Systems*** is not null,

return result C

else if ***NOXE System ID Array*** for the location is not null,

For each SystemID in the ***NOXE System ID Array*** for the location:

Locate the latest ***AppendixETestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the

SystemID being checked,

If found,
Set *ExpirationDate* to five years after the end of the quarter of the ***AppendixETestRecordsByLocationForQAStatus***.EndDate.

If *ExpirationDate* is prior to the current calendar date,
Append the SystemIdentifier being checked to ***Expired Systems***.

else if *ExpirationDate* is on or before the last day of the ***Current Reporting Period***,
Set Expiration Text to "will be expiring at the end of the reporting period".
Append the SystemIdentifier being checked to ***Expired Systems***.

else if *ExpirationDate* is on or before the last day of the quarter following the ***Current Reporting Period***,
Append the SystemIdentifier being checked to ***Expiring Systems***.

If both ***Expired Systems*** and ***Expiring Systems*** are not null,
return result D
else if ***Expired Systems*** is not null,
return result E
else if ***Expiring Systems*** is not null,
return result F

else if ***LME Fuel Array*** for the location is not null,

For each FuelCode in the ***LME Fuel Array*** for the location:

Locate the latest ***UnitDefaultTestRecordsByLocationForQAStatus*** for the location where the FuelCode is equal to the FuelCode being checked,

If found,
Set *ExpirationDate* to five years after the end of the quarter of the ***UnitDefaultTestRecordsByLocationForQAStatus***.EndDate.

If *ExpirationDate* is prior to the current calendar date,
Append the FuelCode being checked to ***Expired Systems***.

else if *ExpirationDate* is on or before the last day of the ***Current Reporting Period***,
Set Expiration Text to "will be expiring at the end of the reporting period".
Append the FuelCode being checked to ***Expired Systems***.

else if *ExpirationDat*e is on or before the last day of the quarter following the ***Current Reporting Period***,
Append the FuelCode being checked to ***Expiring Systems***.

If both ***Expired Systems*** and ***Expiring Systems*** are not null,
return result G
else if ***Expired Systems*** is not null,
return result H
else if ***Expiring Systems*** is not null,
return result I

Results:

Result	Response	Severity
A	Warning: The three-level RATA conducted for each of the following monitoring systems [TEXT]: System ID(s) [EXPIRED]. Except for a 720 operating-hour grace period extension, you will need monitor stack flow with another FLOW system or report substitute data until you perform another three-level RATA. In addition, the three-level RATA conducted for each of the following monitoring systems will expire at the end of the next reporting period: System ID(s) [EXPIRING]. Three-level RATAs expire after five years.	Informational Message
B	Warning: The three-level RATA conducted for each of the following monitoring systems [TEXT]: System ID(s) [EXPIRED]. Except for a 720 operating-hour grace period extension, you will need monitor stack flow with another FLOW system or report substitute data until you perform another three-level RATA. Three-level RATAs expire after five years.	Informational Message
C	Prior Notice: The three-level RATA conducted for each of the following monitoring systems will expire at the end of the next reporting period: System ID(s) [EXPIRING]. Three-level RATAs expire after five years.	Informational Message
D	Warning: The Appendix E test conducted to determine the NOx correlation curve for each of the following monitoring systems [TEXT]: System ID(s) [EXPIRED]. In subsequent reporting periods, you will need report substitute data until you perform another Appendix E test. In addition, the Appendix E test conducted for each of the following monitoring systems will expire at the end of the next reporting period: System ID(s) [EXPIRING]. Appendix E tests expire after five years.	Informational Message
E	Warning: The Appendix E test conducted to determine the NOx correlation curve for each of the following monitoring systems [TEXT]: System ID(s) [EXPIRED]. In subsequent reporting periods, you will need report substitute data until you perform another Appendix E test. Appendix E tests expire after five years.	Informational Message
F	Prior Notice: The Appendix E test conducted to determine the NOx correlation curve for each of the following monitoring systems will expire at the end of the next reporting period: System ID(s) [EXPIRING]. Appendix E tests expire after five years.	Informational Message
G	Warning: The LME Unit Default Test conducted to determine the default NOx emission rate for each of the following fuels [TEXT]: Fuel Code(s) [EXPIRED]. In subsequent reporting periods, you will need report substitute data until you perform another Unit Default Test. In addition, the Unit Default Test conducted for each of the following fuels will expire at the end of the next reporting period: Fuel Code(s) [EXPIRING]. LME Unit Default Tests expire after five years.	Informational Message
H	Warning: The LME Unit Default Test conducted to determine the default NOx emission rate for each of the following fuels [TEXT]: Fuel Code(s) [EXPIRED]. In subsequent reporting periods, you will need report substitute data until you perform another Unit Default Test. LME Unit Default Tests expire after five years.	Informational Message
I	Prior Notice: The LME Unit Default Test conducted to determine the default NOx emission rate for each of the following fuels will expire at the end of the next reporting period: Fuel Code(s) [EXPIRING]. LME Unit Default Tests expire after five years.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURGEN-16

Check Name: Ignored Offline Daily Interference Check

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

If (***Ignored Daily Interference Tests*** == true)
return result A.

Results:

Result	Response	Severity
A	You reported one or more daily interference checks that will not fulfill your daily testing requirement for your stack flow monitors, because these tests were performed while the unit was not operating. These tests have been assigned a CalculatedTestResult of "IGNORED". They can be viewed on the Other Daily Tests tab of the View Detailed Emissions Screen.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURGEN-17

Check Name: Missing Peaking or Gas Fired Qualification Percent Check

Related Former Checks:

Applicability: General Check

Description: This check determines whether qualification percent rows are missing for a Year Round or Ozone Season Monitor Qualification record.

Specifications:

If ***MpSuccessfullyEvaluated*** equals true,

Set ***QualificationPercentMissingList*** = null.

For each *QualificationRecord* record in ***MpQualificationRecords*** where QualificationTypeCode is equal to "PK", "SK" or "GF", BeginDate is on or before ***CurrentReportingPeriodEndHour***, and EndDate is null or or is on or after ***CurrentReportingPeriodBeginHour***:

Locate the *QualificationPercentRecord* record in ***MpQualificationPercentRecords*** where MonitorQualificationId is equal to *QualificationRecord*.MonitorQualificationId, and QualificationDataYear is equal to ***CurrentReportingPeriodYear***.

If not found,

If *QualificationRecord*.QualificationTypeCode is equal to "GF",

Add "Gas-Fired Unit " concatenated with *QualificationRecord*.LocationId to ***QualificationPercentMissingList***.

Else if *QualificationRecord*.QualificationTypeCode is equal to "PK",

Add "Year-Round Peaking Unit " concatenated with *QualificationRecord*.LocationId to ***QualificationPercentMissingList***.

Else if *QualificationRecord*.QualificationTypeCode is equal to "SK",

Add "Ozone-Season Peaking Unit " concatenated with *QualificationRecord*.LocationId to ***QualificationPercentMissingList***.

If ***QualificationPercentMissingList*** is not null,
return result A.

Results:

Result	Response	Severity
A	You did not report a current year percent record for [QualPctMissingList].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-18

Check Name: Validate Unit Fuel

Related Former Checks:

Applicability:

Description: Validate that the fuel designated as primary is used at least 60 percent for the entire year of the time for a unit which has operated greater 168 op hours

Specifications:

If Quarter of the ***Current Reporting Period*** is equal to 1,

For each unit in ***MonitoringPlanLocationRecords***

Locate all ***Facility Unit Fuel Records*** for the unit where :

1) Indicator Code is equal to "P"
2) BeginDate is prior to the end date of ***Current Reporting Period***.
3) EndDate is null OR is after the begin date of ***Current Reporting Period***.

If found,

Set *Sum of Op Hours* to 0

Locate all ***Facility Operating Supp Data Records*** for the unit where:

a) The calendar year is the year prior to the calendar year of ***Current Reporting Period***.
b) Parameter Code is equal to "OPHOURS".
c) Fuel Code is equal to NULL.

For each located record

Increment *Sum of Op Hours* by ***Facility Operating Supp Data Records***.OpValue

If *Sum of Op Hours* > 168

Set *Sum of Op Hours by Fuel* to null

Locate all ***Facility Operating Supp Data Records*** for the unit where:

a) The calendar year is the year prior to the calendar year of ***Current Reporting Period***.
b) Parameter Code is equal to "OPHOURS".
c) Fuel Code is **NOT** equal to NULL.

For each located record

Increment *Sum of Op Hours by Fuel* for ***Facility Operating Supp Data Records***.FuelCode by ***Facility Operating Supp Data Records***.OpValue

For each FuelCode in *Sum of Op Hours by Fuel*

If (*Sum of Op Hours by Fuel* / *Sum of Op Hours* is greater than or equal to 0.60)

For each located ***Facility Unit Fuel Records*** record

If ***Facility Unit Fuel Records***.FuelCode is not equal to the FuelCode for *Sum of Op Hours by Fuel*

return A

Results:

Result	Response	Severity
A	The current active primary fuel type defined in the monitoring plan is inconsistent with the prior year operating hours by fuel type. Please update the primary fuel type in the monitoring plan to match the prior year predominant fuel type by operating hours.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-19

Check Name: Initialize Sorbent Trap Check Parameters

Related Former Checks:

Applicability:

Description: Initializes Sorbent Trap Dictionary and Sorbent Trap Record. Sorbent Trap Record is always initialized to null, but is initialized here so that it always exists and is owned by an ancestor category.

Specifications:

Initialize ***MatsSorbentTrapDictionary*** to an empty dictionary.
Initialize ***MatsSamplingTrainDictionary*** to an empty dictionary.
Initialize ***MatsSorbentTrapListByLocationArray*** with the number of elements equal to ***CurrentLocationCount***.

Set ***MatsSorbentTrapEvaluationNeeded*** to false.

If Count of records in ***MatsSorbentTrapRecords*** where SupplementalDataIndicator is false is greater than 0,

Set ***MatsSorbentTrapEvaluationNeeded*** to true.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-20

Check Name: Initialize Weekly System Integrity Test Dictionary

Related Former Checks:

Applicability:

Description: Initializes the weekly system integrity test dictionary to have a ComponentId key and an entry with the following fields:

1) MostRecentTestRecord to hold a CurrentWeeklySystemIntegrity record.
2) LastEvaluatedTestRecord to hold a CurrentWeeklySystemIntegrity record.
3) OpertingDateList to hold a date list.
4) LastOperatingDate to hold a date.

Specifications:

Initialize ***WsiTestDictionary*** with a string key for ComponentId, and an entry with the following fields:

1) MostRecentTestRecord to hold a ***CurrentWeeklySystemIntegrity*** record.
2) LastEvaluatedTestRecord to hold a ***CurrentWeeklySystemIntegrity*** record.
3) OperatingDateList to hold a date list.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-21

Check Name: Initialize General Information

Related Former Checks:

Applicability: General Check

Description: Initializes list used during the evaluation process by individual checks.

Validation Tables:

Test Result Code (Lookup Table)

Specifications:

Append each ***TestResultCodeLookupTable***.TestResultCode to ***TestResultCodeList*** delimited by a comma.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-22

Check Name: Initialize System Parameters

Related Former Checks:

Applicability: General Check

Description: Initializes values based on system parameters.

Validation Tables:

Vw System Parameter (Lookup Table)

Specifications:

Set ***MatsDailyCalRequiredDate*** to null.

Locate ***SystemParameterLookupTable*** record where Sys_Param_Name is equal to 'MATS_RULE.

If found,

Set ***MatsDailyCalRequiredDate*** to the located ***SystemParameterLookupTable***.Param_Value2.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-23

Check Name: Initialize Program Lists

Related Former Checks:

Applicability: General Check

Description: Initializes program code lists that contain programs that:

1) Are ozone season programs
2) Use RUEs.
3) Require SO2 System certification.
4) Require NOX System certification.
5) Require NOX Concentration System certification.

Specifications:

Set ***ProgramIsOzoneSeasonList*** to "".
Set ***ProgramRequiresNoxSystemCertificationList*** to "".
Set ***ProgramRequiresNoxcSystemCertificationList*** to "".
Set ***ProgramRequiresSo2SystemCertificationList*** to "".
Set ***ProgramUsesRueList*** to "".

For each *ProgramCodeRow* in ***ProgramCodeTable***,

If *ProgramCodeRow*.OzoneSeasonIndicator is equal to 1,

Append *ProgramCodeRow*.ProgramCode to ***ProgramIsOzoneSeasonList***.

If *ProgramCodeRow*.NoxCertificationRequiredIndicator is equal to 1,

Append *ProgramCodeRow*.ProgramCode to ***ProgramRequiresNoxSystemCertificationList***.

If *ProgramCodeRow*.NoxcCertificationRequiredIndicator is equal to 1,

Append *ProgramCodeRow*.ProgramCode to ***ProgramRequiresNoxcSystemCertificationList***.

If *ProgramCodeRow*.So2CertificationRequiredIndicator is equal to 1,

Append *ProgramCodeRow*.ProgramCode to ***ProgramRequiresSo2SystemCertificationList***.

If *ProgramCodeRow*.UsesRueIndicator is equal to 1,

Append *ProgramCodeRow*.ProgramCode to ***ProgramUsesRueList*** .

Results:

Result	Response	Severity

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-24

Check Name: Store Constant Values

Related Former Checks:

Applicability: General Check

Description: This check stores values that will not change during the evaluation session, but are not readily accessible.

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

Set ***MwLoadHourlyTolerance*** to Tolerance from the ***HourlyEmissionsTolerancesCrossCheckTable*** record where Parameter equals "LOAD" and UOM equals "MW".

Initialize the number of elements in ***LocationNameArray*** to the number of records in ***MonitorPlanLocationRecords***.
Set each element in ***LocationNameArray*** to either LocationName in the corresponding record in ***MonitorPlanLocationRecords***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-25

Check Name: Return Results for Invalid Daily Calibration Cylinder Ids

Related Former Checks:

Applicability: General Check

Description: Returns a result when invalid cylinder ids were encountered during daily calibration evaluation.

Specifications:

Set ***FormattedCylinderIdList*** to "".

When ***InvalidCylinderIdList*** is not null and contains ids,

Append each id in ***InvalidCylinderIdList*** to ***FormattedCylinderIdList***.
Return result A.

Results:

Result	Response	Severity
A	The format is invalid for daily calibration cylinder id(s) [CylinderIdList]. A cylinder id can only contain alphanumeric capitalized characters, hyphens, and periods.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Evaluation

Check Code: HOURGEN-26

Check Name: Initialize QA Certification Event Supplemental Data Storage

Related Former Checks:

Applicability: General Check

Description: Initializes the storage for QA Certification Event Supplemental Data.

Specifications:

Initialize ***QaCertEventSuppDataDictionaryArray*** to an array with the number of elements matching the number of records in ***MonitoringPlanLocationRecords***.
Initialize ***QaCertEventSuppDataDictionaryBySystem*** to an empty dictionary.
Initialize ***QaCertEventSuppDataDictionaryByComponent*** to an empty dictionary.

For each *LocationPosition* in ***MonitoringPlanLocationRecords***,

Set *MonitoringPlanLocationRecord* to the record at *LocationPosition.*
Initialize the ***QaCertEventSuppDataDictionaryArray*** element at *LocationPosition* to an empty dictionary.

For each *QaCertEventRecord* in ***QaCertEventsForEmEvaluation*** records where:

1) MonitorLocationKey is equal to *MonitoringPlanLocationRecord*.MonitorLocationKey.
2a) QaCertEventDate is between ***CurrentReportingPeriodBeginDate*** and ***CurrentReportingPeriodEndDate***, OR
2b) ConditionalDataBeginDate is between ***CurrentReportingPeriodBeginDate*** and ***CurrentReportingPeriodEndDate***.

Perform the following for both *QaCertEventRecord*.QaCertEventDatehour and *QaCertEventRecord*.ConditionalDataBeginDatehour.

For *QaCertEventRecord*.QaCertEventDatehour :

1) Set *TargetDatehourCode* = "QaCertEventDate".
2) Set *TargetDatehourValue* to *QaCertEventRecord*.QaCertEventDatehour
3) Set *TimeType* to "Date".
4) Note that QaCertEventDatehour will never have a null value.

For *QaCertEventRecord*.ConditionalDataBeginDatehour:

1) Set *TargetDatehourCode* = "ConditionalDataBeginHour".
2) Set *TargetDatehourValue* to *QaCertEventRecord*.ConditionalDataBeginDatehour
3) Set *TimeType* to "Hour".
4) Note that ConditionalDataBeginDatehour may have a null value.

If *TargetDatehourValue* is not null AND the quarter of *TargetDatehourValue* matches ***CurrentReportingPeriodYear***/***CurrentReportingPeriodQuarter***,

Create *QaCertEventSuppDataRecord* with:

1) QaCertEventKey set to *QaCertEventRecord*.QaCertEventKey.
2) MonitorLocationKey set to *QaCertEventRecord*.MonitorLocationKey.
3) ReportingPeriodKey set to ***CurrentReportingPeriod***.
4) TargetDatehourCode set to *TargetDatehourCode.*
5) TargetDatehourValue set to *TargetDatehourValue*.
6) TimeType set to *TimeType.*
7) OpCount set to 0.
8) SystemQualityAssuredCount set to 0 if *QaCertEventRecord*.MonSysKey is not null, otherwise

set to null.
9) ComponentQualityAssuredCount set to 0 if *QaCertEventRecord*.ComponentKey is not null, otherwise set to null.
10) MayAndJuneOpCount set to 0 if ***CurrentReportingPeriodQuarter*** is equal to 2, otherwise set to null.
11) MayAndJuneSystemQualityAssuredCount set to 0 if ***CurrentReportingPeriodQuarter*** is equal to 2 AND *QaCertEventRecord*.MonSysKey is not null, otherwise set to null.
12) MayAndJuneComponentQualityAssuredCount set to 0 ***CurrentReportingPeriodQuarter*** is equal to 2 AND *QaCertEventRecord*.ComponentKey is not null, otherwise set to null.

Add *QaCertEventSuppDataRecord* to the dictionary at *LocationPosition* in ***QaCertEventSuppDataDictionaryArray*** using *QaCertEventRecord* .QaCertEventKey and *TargetDatehourCode* as the key.

If *QaCertEventRecord*.MonSysKey is not null,
If ***QaCertEventSuppDataDictionaryBySystem*** does not contain a key equal to *QaCertEventRecord*.MonSysKey ,
Add *QaCertEventRecord*.MonSysKey to ***QaCertEventSuppDataDictionaryBySystem*** with an empty list.

Add *QaCertEventSuppDataRecord* to the list in ***QaCertEventSuppDataDictionaryBySystem*** for *QaCertEventRecord*.MonSysKey ,

If *QaCertEventRecord*.ComponentKey is not null,
If ***QaCertEventSuppDataDictionaryByComponent*** does not contain a key equal to *QaCertEventRecord*.ComponentKey,
Add *QaCertEventRecord*.ComponentKey to ***QaCertEventSuppDataDictionaryByComponent*** with an empty list.

Add *QaCertEventSuppDataRecord* to the list in ***QaCertEventSuppDataDictionaryByComponent*** for *QaCertEventRecord*.ComponentKey ,

Results:

Result	Response	Severity

Usage:

1 Process/Category: Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-27

Check Name: Initialize System Operating Supplemental Data Storage

Related Former Checks:

Applicability: General Check

Description: Initializes the storage for System Operating Supplemental Data.

Specifications:

Initialize ***SystemOperatingSuppDataDictionaryArray*** to an array with the number of elements matching the number of records in ***MonitoringPlanLocationRecords***.

For each *LocationPosition* in ***MonitoringPlanLocationRecords***,

Set *MonitoringPlanLocationRecord* to the record at *LocationPosition.*
Initialize the ***SystemOperatingSuppDataDictionaryArray*** element at *LocationPosition* to an empty dictionary.

For each *MonitorSystemRecord* in ***MonitorSystemsForEmEvaluation*** records where:

1) MonitorLocationID is equal to *MonitoringPlanLocationRecord*.MonitorLocationID.
2) SystemTypeCode is in the set { "CO2", "FLOW", "GAS", "H2O", "H2OM", "H2OT", "HCL", "HF", "HG", "NOX", "NOXC", "NOXE", "NOXP", "O2", "SO2", "ST", "OILM", "OILV" }.
3) BeginDate is on or before ***CurrentReportingPeriodEndDate***.
4) EndDate is null OR is on or after ***CurrentReportingPeriodBeginDate***.

If *SupplementalDataDictionary* does NOT contain key *MonitorSystemRecord*.MonitoringSystemID,

Create *SystemOperatingtSuppDataRecord* with:

1) MonitoringSystemID set to *MonitorSystemRecord*.MonitoringSystemID.
2) ReportingPeriodID set to ***CurrentReportingPeriod***.
3) MonitorLocationID set to *MonitorSystemRecord*.MonitoringLocationID
4) OpDays set to 0.
5) OsDays set to 0.
6) OpHours set to 0.
7) OsHours set to 0.

Add a *SystemOperatingtSuppDataRecord* to the dictionary at *LocationPosition* in ***SystemOperatingSuppDataDictionaryArray*** using *SystemOperatingtSuppDataRecord*.MonitoringSystemID as the key.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-28

Check Name: Initialize Component Operating Supplemental Data Storage

Related Former Checks:

Applicability: General Check

Description: Initializes the storage for Component Operating Supplemental Data.

Specifications:

Initialize ***ComponentOperatingSuppDataDictionaryArray*** to an array with the number of elements matching the number of records in ***MonitoringPlanLocationRecords***.

For each *LocationPosition* in ***MonitoringPlanLocationRecords***,

Set *MonitoringPlanLocationRecord* to the record at *LocationPosition.*

Set *SupplementalDataDictionary* to an empty dictionary.
Initialize the ***ComponentOperatingSuppDataDictionaryArray*** element at *LocationPosition* to *SupplementalDataDictionary* .

For each *MonitorSystemComponentRecord* in ***MonitorSystemComponentsForEmEvaluation*** records where:

1) MonitorLocationID is equal to *MonitoringPlanLocationRecord*.MonitorLocationID.
2) ComponentTypeCode is in the set { "CO2", "FLOW", "HCL", "HF", "HG", "H2O", "NOX", "O2", "SO2" }.
3) BeginDate is on or before ***CurrentReportingPeriodEndDate***.
4) EndDate is null OR is on or after ***CurrentReportingPeriodBeginDate***.

If *SupplementalDataDictionary* does NOT contain key *MonitorSystemComponentRecord*.ComponentID,

Create *ComponentOperatingtSuppDataRecord* with:

1) ComponentID set to *MonitorSystemComponentRecord*.ComponentID .
2) ReportingPeriodID set to ***CurrentReportingPeriod***.
3) MonitorLocationID set to *MonitorSystemComponentRecord*.MonitorLocationID.
4) OpDays set to 0.
5) OsDays set to 0.
6) OpHours set to 0.
7) OsHours set to 0.

Add a *ComponentOperatingtSuppDataRecord* to *SupplementalDataDictionary* using *ComponentOperatingtSuppDataRecord*.ComponentID as the key.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-29

Check Name: Initialize Last Quality Assured Value Supplemental Data Storage

Related Former Checks:

Applicability: General Check

Description: Initializes the storage for Last Quality Assured Value Supplemental Data.

Specifications:

Initialize ***LastQualityAssuredValueSuppDataDictionaryArray*** to an array with the number of elements matching the number of records in ***MonitoringPlanLocationRecords***.

For each *LastQualityAssuredValueSuppDataDictionary* in ***LastQualityAssuredValueSuppDataDictionaryArray***,

Initialize the *LastQualityAssuredValueSuppDataDictionary* to an empty dictionary.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-30

Check Name: Initialize Primary Bypass Information

Related Former Checks:

Applicability: General Check

Description: Initializes check parameters used to indicate whether processing primary bypass stacks implemented as systems should occur, and initializes information needed by that processing.

Specifications:

Set ***PrimaryBypassActiveInQuarter*** to false.

Locate records in ***MonitorSystemsForEmEvaluation*** where:

1) SystemDesignationCode equals "PB".
2) BeginDate is on or before ***CurrentReportingPeriodEndDate***.
3) EndDate is on or after ***CurrentReportingPeriodBeginDate.***

If found,

Set ***PrimaryBypassActiveInQuarter*** to true.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-31

Check Name: Initialize No- Op Linearity Checking

Related Former Checks:

Applicability: General Check

Description: Initializes the check parameters used to aid the checking for linearity test that occurred with begin or end hours during non-op hours.

Specifications:

Initialize ***LinearityExistsLocationArray*** as a boolean array with an element for each row in ***MonitoringPlanLocationRecords***.

For each *LocationPosition* in ***MonitoringPlanLocationRecords***,

Set *LocationKey* to the MonitorLocationKey in the record at *LocationPosition* in ***MonitoringPlanLocationRecords***.

When ***LinearityTestRecordsByLocationForQaStatus*** records exist where the MonitorLocationKey equals *LocationKey*.

Set ***LinearityExistsLocationArray*** at *LocationPostion* to true.

Else

Set ***LinearityExistsLocationArray*** at *LocationPostion* to false.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: HOURGEN-4

Check Name: Emission Comment Reporting Period Valid

Related Former Checks:

Applicability: General Check

Description: This check determines whether or not Emission Comment Reporting Period is valid.

Specifications:

For a Emission Comment record:

If ReportingPeriod is null,
return result A.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Fatal

Usage:

1 Process/Category: Emissions Data Entry Screen Evaluation Emission Comments Evaluation

Check Code: HOURGEN-5

Check Name: Submission Comment Valid

Related Former Checks:

Applicability: General Check

Description:

Specifications:

For the Emission Comment record:

If SubmissionComment is null,
return result A.

Results:

Result	Response	Severity
A	You did not provide [fieldname], which is required for [key].	Fatal

Usage:

1 Process/Category: Emissions Data Entry Screen Evaluation Emission Comments Evaluation

Check Code: HOURGEN-6

Check Name: Duplicate Emission Comment Records

Related Former Checks:

Applicability: General Check

Description: This check determines if there is another EmissionComment record with the same key fields.

Specifications:

For a Emission Comment record:

Locate another EmissionComment record for the monitoring plan with a ReportingPeriod equal to the ReportingPeriod in the current record.

If found,
return result A.

Results:

Result	Response	Severity
A	Another [recordtype] record already exists with the same [fieldnames].	Fatal

Usage:

1 Process/Category: Emissions Data Entry Screen Evaluation Emission Comments Evaluation

Check Category:

Hourly Monitor Data

Check Code: HOURMHV-1

Check Name: Initialize SO2C Hourly Monitor Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameters and output parameters for subsequent monitor hourly checks for SO2C.

Specifications:

Current MHV Parameter = "SO2C"
SO2C Calculated Adjusted Value = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-2

Check Name: Initialize H2O Hourly Monitor Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameter and output parameter for subsequent monitor hourly checks for H2O.

Specifications:

Current MHV Parameter = "H2O"
H2O MHV Calculated Adjusted Value = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation

Check Code: HOURMHV-3

Check Name: Initialize NOXC Hourly Monitor Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameter and output parameter for subsequent monitor hourly checks for NOXC.

Specifications:

Current MHV Parameter = "NOXC"
NOXC Calculated Adjusted Value = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation

Check Code: HOURMHV-4

Check Name: Initialize Flow Hourly Monitor Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameters and output parameters for subsequent monitor hourly checks for FLOW.

Specifications:

Current MHV Parameter = "FLOW"
FLOW Calculated Adjusted Value = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation

Check Code: HOURMHV-5

Check Name: Initialize CO2C Hourly Monitor Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameter and output parameter for subsequent monitor hourly checks for CO2C.

Specifications:

Current MHV Parameter = "CO2C"
CO2C MHV Calculated Adjusted Value = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation

Check Code: HOURMHV-6

Check Name: Initialize O2 Dry Hourly Monitor Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameter and output parameter for subsequent monitor hourly checks for O2 Dry.

Specifications:

Current MHV Parameter = "O2D"
O2 Dry Calculated Adjusted Value = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation

Check Code: HOURMHV-7

Check Name: Initialize O2 Wet Hourly Monitor Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameter and output parameter for subsequent monitor hourly checks for O2C Wet.

Specifications:

Current MHV Parameter = "O2W"
O2 Wet Calculated Adjusted Value = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation

Check Code: HOURMHV-8

Check Name: Check MODC in MHV Record

Related Former Checks:

Applicability: CEM Check

Description: Basic check to ensure that MODC reported in the MHV record is valid for the parameter. Also initializes variables for the category.

Specifications:

Monitor Hourly Modc Status = false
Current MHV Parameter Description = ***Current MHV Parameter***
Complete MHV Record Needed = true

case (***Current MHV Parameter***)

SO2C: ***Current MHV Record = Current SO2 Monitor Hourly Record***
Current MHV Component Type = 'SO2'
Current MHV System Type = 'SO2'
Current MHV Default Parameter = 'SO2X'

If (***Current MHV Record.***ModcCode == "23")
If (***SO2 Bypass Code*** == "BYMAXFS")
Current MHV Fuel Specific Hour = true
else
Current MHV Fuel Specific Hour = false

else if (***SO2 Fuel Specific Missing Data*** == true)
Current MHV Fuel Specific Hour = true
else
Current MHV Fuel Specific Hour = false

If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 53, 54, 55})
return result A
Else
Monitor Hourly Modc Status = true

NOXC: ***Current MHV Record = Current NOx Conc Monitor Hourly Record***
Current MHV System Type = 'NOXC'
Current MHV Component Type = 'NOX'
Current MHV Default Parameter = 'NOCX'

NOx Conc MODC = null

If (***Current MHV Record.***ModcCode in set {23, 24})
If (***NOx Mass Bypass Code*** == "BYMAXFS")
Current MHV Fuel Specific Hour = true
else
Current MHV Fuel Specific Hour = false
else if (***NOx Mass Fuel Specific Missing Data*** == true)
Current MHV Fuel Specific Hour = true
else
Current MHV Fuel Specific Hour = false

If (***NOx Conc Needed for Nox Mass Calc*** == true)
If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 15, 17, 18, 19, 20, 21, 22, 23, 24, 53, 54, 55})

return result A
Else
Monitor Hourly Modc Status = true
Else
Complete MHV Record Needed = false

If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 17, 18, 19, 20, 21, 22, 53, 54})

If (***Current MHV Record.***ModcCode == 46)
Monitor Hourly Modc Status = true
Else
return result B
Else
Monitor Hourly Modc Status = true
NOx Conc MODC = ***Current MHV Record.***ModcCode

FLOW: ***Current MHV Record = Current Stack Flow Hourly Record***
Current MHV Component Type = 'FLOW'
Current MHV System Type = 'FLOW'
Current MHV Default Parameter = 'FLOX'

If (***SO2 Fuel Specific Missing Data*** == true OR ***CO2 Fuel Specific Missing Data*** == true OR ***NOx Mass Fuel Specific Missing Data*** == true OR ***Heat Input Fuel Specific Missing Data*** == true)
Current MHV Fuel Specific Hour = true
else
Current MHV Fuel Specific Hour = false

If (***FlowNeededForPart75*** is true)

If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 20, 53, 54, 55})
return result A
Else
Monitor Hourly Modc Status = true

Else /* Flow Needed for MATS */

If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 20, 46, 53, 54})
return result F
Else
Monitor Hourly Modc Status = true

CO2C: ***Current MHV Record = Current CO2 Conc Monitor Hourly Record***
Current MHV Component Type = 'CO2'
Current MHV System Type = 'CO2'
Current MHV Default Parameter = 'CO2X'

CO2C MHV MODC = ***Current CO2 Conc Monitor Hourly Record***.ModcCode

If ((***CO2 Fuel Specific Missing Data*** == true AND ***CO2 Conc Checks Needed for CO2 Mass Calc*** == true) OR (***Heat Input Fuel Specific Missing Data*** == true AND ***CO2 Conc Checks Needed for Heat Input*** == true))
Current MHV Fuel Specific Hour = true
else
Current MHV Fuel Specific Hour = false

If ((***CO2 Conc Checks Needed for Heat Input*** == true) OR (***CO2 Conc Checks Needed for CO2 Mass Calc*** == true))
If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 06, 07, 08, 09, 10, 12, 17, 20, 21, 23, 53, 54, 55})

return result A
Else
Monitor Hourly Modc Status = true

If (***Current CO2 Conc Missing Data Monitor Hourly Record*** is not null)
If ((***CO2 Diluent Checks Needed for NOx Rate Calc*** == true) OR (***CO2 Diluent Needed for MATS*** == true)) AND (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 17, 20, 21, 53, 54})
return result E
Else
Complete MHV Record Needed = false

If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 17, 20, 21, 53, 54})

If (***Current MHV Record.***ModcCode == 46)
Monitor Hourly Modc Status = true
Else
return result C
Else
Monitor Hourly Modc Status = true

O2D: ***Current MHV Record = Current O2 Dry Monitor Hourly Record***
Current MHV Component Type = 'O2'
Current MHV System Type = null
Current MHV Default Parameter = 'O2N'

O2 Dry MODC = ***Current O2 Dry Monitor Hourly Record***.ModcCode

If (***Current MHV Record***.MoistureBasis is null)
Current MHV Parameter Description = "O2C"
else
Current MHV Parameter Description = "O2C with a MoistureBasis of " + ***Current MHV Record***.MoistureBasis

If (***Heat Input Fuel Specific Missing Data*** == true AND ***O2 Dry Checks Needed for Heat Input*** == true)
Current MHV Fuel Specific Hour = true
else
Current MHV Fuel Specific Hour = false

If (***O2 Dry Checks Needed for Heat Input*** == true)
If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 06, 07, 08, 09, 10, 12, 17, 20, 53, 54, 55})
return result A
Else
Monitor Hourly Modc Status = true

If (***Current O2 Dry Missing Data Monitor Hourly Record*** is not null)
If (((***O2 Dry Checks Needed for NOx Rate Calc*** == true) OR (***O2 Dry Needed for MATS*** == true)) AND (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 17, 20, 53, 54}))
return result E

Else
Complete MHV Record Needed = false

If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 17, 20, 53, 54})

If (***Current MHV Record.***ModcCode == 46)
Monitor Hourly Modc Status = true

Else
return result D
Else
Monitor Hourly Modc Status = true

O2W: ***Current MHV Record = Current O2 Wet Monitor Hourly Record***
Current MHV Component Type = 'O2'
Current MHV System Type = null
Current MHV Default Parameter = 'O2N'

O2 Wet MODC = ***Current O2 Wet Monitor Hourly Record***.ModcCode

If (***Current MHV Record***.MoistureBasis is null)
Current MHV Parameter Description = "O2C"
else
Current MHV Parameter Description = "O2C with a MoistureBasis of " + ***Current MHV Record***.MoistureBasis

If (***Heat Input Fuel Specific Missing Data*** == true AND ***O2 Wet Checks Needed for Heat Input*** == true)
Current MHV Fuel Specific Hour = true
else
Current MHV Fuel Specific Hour = false

If (***O2 Wet Checks Needed for Heat Input*** == true)
If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 06, 07, 08, 09, 10, 12, 17, 20, 53, 54, 55})
return result A
Else
Monitor Hourly Modc Status = true

If (***Current O2 Wet Missing Data Monitor Hourly Record*** is not null)
If (((***O2 Wet Checks Needed for NOx Rate Calc*** == true) OR (***O2 Wet Needed for MATS*** == true)) AND (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 17, 20, 53, 54}))
return result E

Else
Complete MHV Record Needed = false

If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 17, 20, 53, 54})

If (***Current MHV Record.***ModcCode == 46)
Monitor Hourly Modc Status = true
Else
return result D
Else
Monitor Hourly Modc Status = true

H2O: ***Current MHV Record = Current H2O Monitor Hourly Record***
Current MHV Parameter = 'H2O'

H2O MHV MODC = ***Current H2O Monitor Hourly Record***.ModcCode

If (***H2O Method Code*** == "MMS")
Current MHV Component Type = "H2O"
else
Current MHV Component Type = "DAHS"

Current MHV System Type = null

Current MHV Default Parameter = null'

If (***H2O Fuel Specific Missing Data*** == true)
Current MHV Fuel Specific Hour = true
else
Current MHV Fuel Specific Hour = false

If (***Current MHV Record.***ModcCode not in set {01, 02, 03, 04, 06, 07, 08, 09, 10, 12, 21, 53, 54, 55})
return result A
else
Monitor Hourly Modc Status = true

CO2CSD:
Current MHV Record = Current CO2 Conc Missing Data Monitor Hourly Record
Current MHV Component Type = 'CO2'
Current MHV System Type = 'CO2'
Current MHV Default Parameter = 'CO2X'
Current MHV Parameter Description = "CO2C (Substitute Data)"

If ((***CO2 Fuel Specific Missing Data*** == true AND ***CO2 Conc Checks Needed for CO2 Mass Calc*** == true) OR (***Heat Input Fuel Specific Missing Data*** == true AND ***CO2 Conc Checks Needed for Heat Input*** == true))
Current MHV Fuel Specific Hour = true
else
Current MHV Fuel Specific Hour = false

If (***Current MHV Record.***ModcCode not in set {06, 07, 08, 09, 10, 12, 55})
return result A
Else
Monitor Hourly Modc Status = true

O2CSD:

if (***Current O2 Dry Missing Data Monitor Hourly Record*** is not null)
Current MHV Record = Current O2 Dry Missing Data Monitor Hourly Record
else
Current MHV Record = Current O2 Wet Missing Data Monitor Hourly Record

Current MHV Component Type = 'O2'
Current MHV System Type = null
Current MHV Default Parameter = 'O2N'
Current MHV Parameter Description = "O2C (Substitute Data)"

If (***Heat Input Fuel Specific Missing Data*** == true AND ***O2 Dry Checks Needed for Heat Input*** == true)
Current MHV Fuel Specific Hour = true
else
Current MHV Fuel Specific Hour = false

If (***Current MHV Record.***ModcCode not in set {06, 07, 08, 09, 10, 12, 55})
return result A
Else
Monitor Hourly Modc Status = true

Results:

Result	Response	Severity
A	The MODCCode reported in the MHV record for [param] is invalid.	Critical Error Level 1
B	You reported a MODCCode of [MODC] in the MHV record for NOXC, but this MODC is not appropriate when NOX concentration is used in a NOx-diluent system to calculate the NOx emission rate.	Critical Error Level 1
C	You reported a MODCCode of [MODC] in the MHV record for CO2C, but this MODC is not appropriate when CO2 concentration is only used to calculate a heat input based emission rate.	Critical Error Level 1
D	You reported a MODCCode of [MODC] in the MHV record for [param], but this MODC is not appropriate when O2 concentration is not used to calculate the heat input rate.	Critical Error Level 1
E	You reported a MODCCode of [MODC] in the MHV record for [param], but this MODC is not appropriate when this value is used in a diluent system to calculate the heat input based emission rate.	Critical Error Level 1
F	You reported a MODCCode of [MODC] in the MHV record for [param], but this MODC is not appropriate for MATS-only units.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data
3	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data
9	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-9

Check Name: Check Percent Monitor Availability in MHV Record

Related Former Checks:

Applicability: CEM Check

Description: Performs a series basic checks to ensure that the reported monitor percent available is between 0 and 100, inclusive, then checks to see that percent available is within permitted ranges for specific MODC codes

Specifications:

Monitor Hourly Pma Status = false
Monitor Hourly Missing Data Status = true

```
If (Monitor Hourly Modc Status == true)
    If ( Current MHV Record.PercentAvailable is NULL)
        if (Complete MHV Record Needed == false)
            Monitor Hourly Pma Status = true
        else
            if (Current MHV Record.ModcCode not in set {01, 02, 03, 04, 16, 17, 18, 19, 20, 21, 22, 53, 54} AND Legacy
            Data Evaluation == true)
                Monitor Hourly Pma Status = true
                return result A
            else
                return result B
    else
        if (Complete MHV Record Needed == false)
            return result C

        else if (Current MHV Record.PercentAvailable> 100.0 OR
                Current MHV Record.PercentAvailable < 0.0)
            return result D

        Else
            case ( Current MHV Record.ModcCode )
                = 06:
                If  Current MHVRecord.PercentAvailable >= 90.0
                    Monitor Hourly Pma Status = true
                Else
                    return result E

                = 08:
                If  Current MHV Record.PercentAvailable >= 95.0
                    Monitor Hourly Pma Status = true
                Else
                    return result E

                = 09:
                If  Current MHV Record.PercentAvailable >= 90.0 AND Current MHV Record.PercentAvailable < 95.0
                    Monitor Hourly Pma Status = true
                Else
                    return result E

                = 10:
                If  Current MHV Record.PercentAvailable >=80.0 AND Current MHV Record.PercentAvailable < 90.0
                    Monitor Hourly Pma Status = true
                Else if  Current MHV Parameter in {FLOW,NOXC} and Current MHV Record.PercentAvailable >=
                90.0
                    Monitor Hourly Pma Status = true
```

return result F
Else
return result E

= 11:
If ***Current MHV Record***.PercentAvailable >=90.0
Monitor Hourly Pma Status = true
Else
return result E

All other MODC Codes:
Monitor Hourly Pma Status = true

Results:

Result	Response	Severity
A	You reported an MODCCode of [ModcCode] in the MHV record for [param], but you did not report a value for PercentAvailable. While this is not required for legacy EDR data, it is required in all MHV records for ECMPS.	Informational Message
B	You did not report PercentAvailable in the MHV record for [param].	Critical Error Level 1
C	You reported PercentAvailable in the MHV record for [param], but this value should not be reported when the monitoring system is only being used to calculate the NOX emission rate, moisture, and/or CO2 concentration. In that case, the percent monitor availability should be reported in the appropriate DHV record.	Non-Critical Error
D	The PercentAvailable reported in the MHV record for [param] is invalid. This value must be between 0 and 100.	Critical Error Level 1
E	You reported an MODCCode of [modcCode] in the MHV record for [param], but the PercentAvailable is not appropriate for this MODC.	Critical Error Level 1
F	You reported an MODCCode of 10 in the [type] record for [param], but the PercentAvailability is greater than or equal to 90. When the PMA is greater than or equal to 90, you should only report an MODC of 10 to indicate that you used the maximum hourly value in the lookback period for the next available higher load bin, because there were no quality-assured data in the bin corresponding to the current load range. (See Part 75.33(c)(5).)	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data
3	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data
9	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-10

Check Name: Check Prior QA'd Hours for MODC 07

Related Former Checks:

Applicability: CEM Check

Description: For Method of Determination Code 07, all prior hours in reporting period are checked to ensure that total of QA'd hours is below a certain threshold

Specifications:

```
if (Monitor Hourly Modc Status == true AND Monitor Hourly PMA Status == true)
    if (Current MHV Record.ModcCode == 07)

        if (Current MHV Parameter in set {O2D, O2W, O2CSD})
            Prior QA Hours = count MonitorHourlyValueData records where
                MonitorHourlyValueData.ModcCode in set {01, 02, 04, 17, 20, 53} AND
                MonitorHourlyValueData.ParameterCode = "O2C" AND
                (MonitorHourlyValueData.MoistureBasis = Current MHV Record.MoistureBasis OR
                MonitorHourlyValueData.MoistureBasis is null) AND
                ( MonitorHourlyValueData.BeginDate < Current Date OR
                  (MonitorHourlyValueData.BeginDate = Current Date AND MonitorHourlyValueData.BeginHour <
                  Current Hour))

        else if (Current MHV Parameter == "CO2CSD")
            Prior QA Hours = count MonitorHourlyValueData records where
                MonitorHourlyValueData.ModcCode in set {01, 02, 04, 17, 20, 21, 53} AND
                MonitorHourlyValueData.ParameterCode = "CO2C" AND
                ( MonitorHourlyValueData.BeginDate < Current Date OR
                  (MonitorHourlyValueData.BeginDate = Current Date AND MonitorHourlyValueData.BeginHour <
                  Current Hour))

        else
            case (Current MHV Parameter)

                SO2C:   MODC Set = {01, 02, 04, 16, 17, 19, 20, 21, 22, 53}
                NOXC:   MODC Set = {01, 02, 04, 17, 19, 20, 21, 22, 53}
                CO2C:   MODC Set = {01, 02, 04, 17, 20, 21, 53}
                FLOW:   MODC Set = {01, 02, 04, 20, 53}
                H2O:    MODC Set = {01, 02, 04, 21, 53}

            Prior QA Hours = count MonitorHourlyValueData records where
                MonitorHourlyValueData.ModcCode in set MODC Set AND
                MonitorHourlyValueData.ParameterCode = Current MHV Record.ParameterCode AND
                ( MonitorHourlyValueData.BeginDate < Current Date OR
                  (MonitorHourlyValueData.BeginDate = Current Date AND MonitorHourlyValueData.BeginHour <
                  Current Hour))

        if (Current MHV Parameter in set {NOXC, FLOW})
            if (Prior QA Hours > 2160)
                Monitor Hourly Missing Data Status = false
                return result A
        else
            if (Prior QA Hours > 720)
                Monitor Hourly Missing Data Status = false
                return result A
```

Results:

Result	Response	Severity
A	You reported an MODCCode of 07 in the MHV record for [param], but too many prior quality assured hours exist in evaluation period for use of this missing data approach.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data
3	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data
9	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-11

Check Name: Check Extraneous Data in MHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that certain fields are null in the MHV record.

Specifications:

Monitor Hourly Null Status = false
Hourly Extraneous Fields = null

if (***Current MHV Record***.AdjustedHourlyValue is not null AND ***Current MHV Parameter*** not in set {SO2C, NOXC, FLOW})
    append "AdjustedHourlyValue" to ***Hourly Extraneous Fields***

if (***Current MHV Record***.MoistureBasis is not null AND ***Current MHV Parameter*** not in set {O2D, O2W, O2CSD})
    append "MoistureBasis" to ***Hourly Extraneous Fields***

if (***Hourly Extraneous Fields*** is not null)
    return result A
else
    Monitor Hourly Null Status = true

Results:

Result	Response	Severity
A	You reported [fieldnames] in the MHV record for [param]. This data should be blank.	Non-Critical Error

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data
3	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data
9	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-12

Check Name: Check For Correct Use of MODCs

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

Current MHV HBHA Value = null

if (***Monitor Hourly Modc Status*** == true AND ***Monitor Hourly PMA Status*** == true)

if (***Current MHV Record***.ModcCode in set {06, 08, 09})

if (***Current MHV Parameter*** in set {O2D, O2W, O2CSD})

Prior MHV Record = latest MonitorHourlyValueData record where
MonitorHourlyValueData.ModcCode in set {01, 02, 03, 04, 17, 20, 53, 54} AND
MonitorHourlyValueData.ParameterCode = "O2C" AND
(MonitorHourlyValueData.MoistureBasis = Current MHV Record.MoistureBasis OR
MonitorHourlyValueData.MoistureBasis is null) AND
[MonitorHourlyValueData.Date < ***Current Date*** OR
(MonitorHourlyValueData.Date = ***Current Date*** AND MonitorHourlyValueData.Hour < ***Current Hour***)]

If *Prior MHV Record* is not null and is in current reporting period

Next MHV Record = earliest MonitorHourlyValueData record where
MonitorHourlyValueData.ModcCode in set {01, 02, 03, 04, 17, 20, 53, 54} AND
MonitorHourlyValueData.ParameterCode = "O2C" AND
(MonitorHourlyValueData.MoistureBasis = Current MHV Record.MoistureBasis OR
MonitorHourlyValueData.MoistureBasis is null) AND
[MonitorHourlyValueData.Date > ***Current Date*** OR
(MonitorHourlyValueData.Date = ***Current Date*** AND MonitorHourlyValueData.Hour > ***Current Hour***)]

If *Next MHV Record* is not null and is in current reporting period

If *Prior MHV Record*.UnadjustedHourlyValue >= 0 AND *Next MHV Record*.UnadjustedHourlyValue >= 0

Current MHV HBHA Value = (*Prior MHV Record*.UnadjustedHourlyValue + *Next MHV Record*.UnadjustedHourlyValue) / 2, ROUNDED to a single decimal.

else

Monitor Hourly Missing Data Status = false
return result A

else

case (***Current MHV Parameter***)

SO2C: *MODC Set* = {01, 02, 03, 04, 16, 17, 19, 20, 21, 22, 53, 54}
NOXC: *MODC Set* = {01, 02, 03, 04, 17, 19, 20, 21, 22, 53, 54}
FLOW: *MODC Set* = {01, 02, 03, 04, 20, 53, 54}
CO2C OR CO2CSD: *MODC Set* = {01, 02, 03, 04, 17, 20, 21, 53, 54}
H2O: *MODC Set* = {01, 02, 03, 04, 21, 53, 54}

if (***Current MHV Parameter*** in set {H2O, CO2C})

Prior Record = latest MonitorHourlyValueData record or DerivedHourlyValueData record where
ParameterCode = ***Current MHV Parameter*** AND
ModcCode in set *MODC Set* AND
(Date < ***Current Date*** OR
(Date = ***Current Date*** AND Hour < ***Current Hour***))

If *Prior Record* is not null and is in current reporting period

Next Record = earliest MonitorHourlyValueData record or DerivedHourlyValueData record where
ParameterCode = ***Current MHV Parameter*** AND
ModcCode in set *MODC Set* AND
(Date > ***Current Date*** OR
(Date = ***Current Date*** AND Hour > ***Current Hour***))

If *Next Record* is not null and is in current reporting period

If *Prior Record*.UnadjustedHourlyValue >= 0 AND *Next Record*.UnadjustedHourlyValue >= 0

Current MHV HBHA Value = (*Prior Record*.UnadjustedHourlyValue + *Next Record*.UnadjustedHourlyValue) / 2, ROUNDED to a single decimal.

else
Monitor Hourly Missing Data Status = false
return result A

else if (***Current MHV Parameter*** == "CO2CSD")

Prior MHV Record = latest MonitorHourlyValueData record where
MonitorHourlyValueData.ParameterCode = "CO2C" AND
MonitorHourValueData.ModcCode in set *MODC Set* AND
[MonitorHourlyValueData.Date < ***Current Date*** OR
(MonitorHourlyValueData.Date = ***Current Date*** AND MonitorHourlyValueData.Hour < ***Current Hour***)]

If *Prior MHV Record* is not null and is in current reporting period

Next MHV Record = earliest MonitorHourlyValueData record where
MonitorHourlyValueData.ParameterCode = "CO2C" AND
MonitorHourValueData.ModcCode in set *MODC Set* AND
[MonitorHourlyValueData.Date > ***Current Date*** OR
(MonitorHourlyValueData.Date = ***Current Date*** AND MonitorHourlyValueData.Hour > ***Current Hour***)]

If *Next MHV Record* is not null and is in current reporting period

If *Prior MHV Record*.UnadjustedHourlyValue >= 0 AND AND *Next MHV Record*.UnadjustedHourlyValue >= 0

Current MHV HBHA Value = (*Prior MHV Record*.UnadjustedHourlyValue + *Next MHV Record*.UnadjustedHourlyValue) / 2, ROUNDED to a single decimal.

else
Monitor Hourly Missing Data Status = false

return result A

else

Prior MHV Record = latest MonitorHourlyValueData record where
MonitorHourlyValueData.ParameterCode = ***Current MHV Parameter*** AND
MonitorHourValueData.ModcCode in set *MODC Set* AND
[MonitorHourlyValueData.Date < ***Current Date*** OR
(MonitorHourlyValueData.Date = ***Current Date*** AND MonitorHourlyValueData.Hour < ***Current Hour***)]

If *Prior MHV Record* is not null and is in current reporting period

Next MHV Record = earliest MonitorHourlyValueData record where
MonitorHourlyValueData.ParameterCode = ***Current MHV Parameter*** AND
MonitorHourValueData.ModcCode in set *MODC Set* AND
[MonitorHourlyValueData.Date > ***Current Date*** OR
(MonitorHourlyValueData.Date = ***Current Date*** AND MonitorHourlyValueData.Hour > ***Current Hour***)]

If *Next MHV Record* is not null and is in current reporting period

If *Prior MHV Record*.AdjustedHourlyValue >= 0 AND *Next MHV Record*.AdjustedHourlyValue >= 0

if (***Current MHV Parameter*** == "FLOW")
Current MHV HBHA Value = (*Prior MHV Record*.AdjustedHourlyValue + *Next MHV Record*.AdjustedHourlyValue) / 2, ROUNDED to the nearest 1000.

else
Current MHV HBHA Value = (*Prior MHV Record*.AdjustedHourlyValue + *Next MHV Record*.AdjustedHourlyValue) / 2, ROUNDED to a single decimal.

else
Monitor Hourly Missing Data Status = false
return result B

else if (***Current MHV Record***.ModcCode == "11")

case (***Current MHV Parameter***)

NOXC: *MODC Set* = {01, 02, 03, 04, 17, 19, 20, 21, 22, 53, 54}
FLOW: *MODC Set* = {01, 02, 03, 04, 20, 53, 54}

Prior Measured MHV Record = MonitorHourlyValueData record at latest time for the location where
MonitorHourlyValueData.ModcCode in set *MODC Set* AND
MonitorHourlyValueData.ParameterCode = ***Current MHV Parameter*** AND
(MonitorHourlyValueData.BeginDate < ***Current Date*** OR
(MonitorHourlyValueData.BeginDate = ***Current Date*** AND MonitorHourlyValueData.BeginHour< ***Current Hour***))

If *Prior Measured MHV Record* is not null and is in the current reporting period
PriorDate = *Prior Measured MHV Record*.BeginDate
PriorHour = *Prior Measured MHV Record*.BeginHour
else
PriorDate = the day prior to the beginning of the current reporting period
PriorHour = 23

Next Measured MHV Record = MonitorHourlyValueData record at earliest time for the location where
MonitorHourlyValueData.ModcCode in set *MODC Set* AND
MonitorHourlyValueData.ParameterCode = ***Current MHV Parameter*** AND
(MonitorHourlyValueData.BeginDate > ***Current Date*** OR
(MonitorHourlyValueData.BeginDate = ***Current Date*** AND MonitorHourlyValueData.BeginHour > ***Current Hour***))

If *Next Measured MHV Record* is not null and is in the current reporting period
NextDate = *Next Measured MHV Record*.BeginDate
NextHour = *Next Measured MHV Record*.BeginHour
else
NextDate = the day after the end of the current reporting period
NextHour = 0

Missing Data Period Length = Count of MonitorHourlyValueData records for the location where
MonitorHourlyValueData.ParameterCode = ***Current MHV Parameter*** AND
(MonitorHourlyValueData.BeginDate > *PriorDate* OR
(MonitorHourlyValueData.BeginDate = *PriorDate* AND MonitorHourlyValueData.BeginHour > *PriorHour*)) AND
(MonitorHourlyValueData.BeginDate < *NextDate* OR
(MonitorHourlyValueData.BeginDate = *NextDate* AND MonitorHourlyValueData.BeginHour < *NextHour*))

if (***Current MHV Record.***PercentAvailable is null OR ***Current MHV Record.***PercentAvailable >= 95.0)
if (*Missing Data Period Length* > 24)
Monitor Hourly Missing Data Status = false
return result C
else
if (*Missing Data Period Length* > 8)
Monitor Hourly Missing Data Status = false
return result C

else if (***Current MHV Record***.ModcCode == "17" AND ***Monitor Hourly System Status*** == true)

Hours of Use of Like Kind Analyzer = Count of MonitorHourlyValueData records for the location and reporting period where
MonitorHourlyValueData.ParameterCode = ***Current MHV Parameter*** AND
MonitorHourlyValueData.ModcCode == "17" AND
(MonitorHourlyValueData.BeginDate < ***Current Date*** OR
(MonitorHourlyValueData.BeginDate = ***Current Date*** AND MonitorHourlyValueData.BeginHour< ***Current Hour***))

If *Hours of Use of Like Kind Analyzer* >= 720

First Use of Like Kind Analyzer Record = MonitorHourlyValueData record at earliest time for the location and and reporting period where
MonitorHourlyValueData.ParameterCode = ***Current MHV Parameter*** AND
MonitorHourlyValueData.ModcCode == "17" AND
(MonitorHourlyValueData.BeginDate < ***Current Date*** OR
(MonitorHourlyValueData.BeginDate = ***Current Date*** AND MonitorHourlyValueData.BeginHour< ***Current Hour***))

If ***Current MHV Record***.MonitoringSystemID is not null,

Locate a ***RATATestRecordsByLocationForQAStatus*** for the location
where the MonitoringSystemID is equal to ***Current MHV Record***.MonitoringSystemID, the

TestResultCode begins with "PASS", and the EndDate/EndHour is after the *First Use of Like Kind Analyzer Record*.Date/Hour and on or prior to the ***Current Date/Current Hour***.

If not found,
return result D

else

Locate all ***Monitor System Component Records for Hour and Location***
where the ComponentID is equal to ***Current MHV Record***.ComponentID.

Locate a ***RATATestRecordsByLocationForQAStatus*** for the location
where the MonitoringSystemID is equal to any MonitoringSystemID in the retrieved Monitor System Component records, the TestResultCode begins with "PASS", and the EndDate/EndHour is after the *First Use of Like Kind Analyzer Record*.Date/Hour and on or prior to the ***Current Date/Current Hour***.

If not found,
return result D

Results:

Result	Response	Severity
A	The UnadjustedHourlyValue reported in the MHV record for [param] either before or after the current hour is invalid.	Critical Error Level 1
B	The AdjustedHourlyValue reported in the MHV record for [param] either before or after the current hour is invalid.	Critical Error Level 1
C	You reported an MODCCode of 11 in the MHV record for [param], but the length of the missing data period exceeds the allowable value for use of this missing data procedure.	Critical Error Level 1
D	You reported an MODCCode of 17 in the MHV record for [param], indicating the use of a like-kind analyzer, but you have used a like-kind analyzer to monitor this parameter for more than 720 hours during this reporting period. You are not allowed to use a like-kind analyzer for more than 720 hours during a calendar year, unless the analyzer is identified as a non-redundant backup and a RATA is performed.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data
3	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data
9	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-13

Check Name: Check System in MHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that a valid Monitoring System is indicated in the MHV record.

Specifications:

Current MHV Mon Sys Record = null
Monitor Hourly System Status = false

If (***Current MHV Parameter*** == "NOXC" AND ***NOx Conc Checks Needed for Nox Mass*** == false)

if (***Current MHV Record***.MonitoringSystemID is NOT null AND ***Legacy Data Evaluation*** == false)
return result A
else
Monitor Hourly System Status = true

else if (***Current MHV Parameter*** == "CO2C" AND ***CO2 Conc Checks Needed for Heat Input*** == false AND ***CO2 Conc Checks Needed for CO2 Mass Calc*** == false AND ***CO2 Diluent Needed for MATS*** == false)

if (***Current MHV Record***.MonitoringSystemID is NOT null AND ***Legacy Data Evaluation*** == false)
return result B
else
Monitor Hourly System Status = true

else if (***Current MHV Parameter*** == "O2W" AND ***O2 Wet Checks Needed for Heat Input*** == false AND ***O2 Wet Checks Needed to Support CO2 Calculation*** == false AND ***O2 Wet Needed for MATS*** == false) OR
(***Current MHV Parameter*** == "O2D" AND ***O2 Dry Checks Needed for Heat Input*** == false AND ***O2 Dry Checks Needed to Support CO2 Calculation*** == false AND ***O2 Dry Needed for MATS*** == false)

if (***Current MHV Record***.MonitoringSystemID is NOT null AND ***Legacy Data Evaluation*** == false)
return result G
else
Monitor Hourly System Status = true

else

If (***Monitor Hourly MODC Status*** == true*)*

if ***Current MHV Record***.MonitoringSystemID is null

case (***Current MHV Parameter***)

SO2C: *MODC Set* = {01, 02, 03, 04, 16, 17, 18, 19, 20, 21, 22}
NOXC: *MODC Set* = {01, 02, 03, 04, 17, 18, 19, 20, 21, 22}
CO2C, O2D, or O2W: *MODC Set* = {01, 02, 03, 04, 17, 18, 20, 21}
FLOW: *MODC Set* = {01, 02, 03, 04, 20}
H2O: *MODC Set* = {01, 02, 03, 04, 21}
CO2CSD or O2CSD: *MODC Set* = {}

if (***Current MHV Record.***ModcCode in set *MODC Set)*
return result C
else
return result H

else

Current MHV Mon Sys Record = find active MonitoringSystemData record for location where MonitoringSystemData.MonitoringSystemID = ***Current MHV Record***.MonitoringSystemID

if ***Current MHV Mon Sys Record*** is null
return result D

else
if (***Current MHV Parameter*** in set {O2D, O2W, O2CSD})

if (***Legacy Data Evaluation*** == true
if (***Current MHV Mon Sys Record***.SystemTypeCode not in set {H2O,O2,CO2,NOXC,NOX})
return result E
else
Monitor Hourly System Status = true
else if (***Current MHV Mon Sys Record***.SystemTypeCode not in {O2, CO2})
return result E
else
Monitor Hourly System Status = true

else if (***Current MHV Parameter*** = "H2O")
if (***Current MHV Mon Sys Record***.SystemTypeCode not in {H2OT, H2OM})
return result E
else
Monitor Hourly System Status = true

else
if (***Current MHV Mon Sys Record***.SystemTypeCode <> ***Current MHV System Type***)
If (***Current MHV Parameter*** in {"CO2C", "CO2CSD"} AND ***Legacy Data Evaluation*** == true AND
Current MHV Mon Sys Record.SystemTypeCode == "NOX")
Monitor Hourly System Status = true
else
return result E
else
Monitor Hourly System Status = true

else
Monitor Hourly System Status = true

Results:

Result	Response	Severity
A	You reported a MonitoringSystemID in the MHV record for NOXC, but this field should be blank when the NOX concentration is used to calculate the NOX emission rate as part of a NOX-diluent system.	Critical Error Level 1
B	You reported a MonitoringSystemID in the MHV record for CO2C, but this field should be left blank when CO2 concentration is not used to calculate CO2 mass, heat input, or a MATS value.	Critical Error Level 1
C	You did not report a MonitoringSystemID in the MHV record for [param]. This information is required when you report measured data.	Critical Error Level 1
D	You reported MonitoringSystemID [ID] in the MHV record for [param], but there is no Monitoring System record for this system in your monitoring plan that was active during the hour.	Critical Error Level 1
E	You reported MonitoringSystemID [ID] in the MHV record for [param], but this SystemTypeCode for this monitoring system is not appropriate.	Critical Error Level 1
F	You reported a MonitoringSystemID in the MHV record for [param], but this is not appropriate when substitute data is used.	Non-Critical Error
G	You reported a MonitoringSystemID in the MHV record for [param], but this field should be left blank when O2 concentration is not used to calculate CO2 concentration, heat input, or a MATS value.	Critical Error Level 1
H	You did not report a MonitoringSystemID in the MHV record for [param]. This information is required when you report for missing data substitution.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data
3	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data
9	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-14

Check Name: Check System Designation Code for System in MHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that the SystemDesignationCode of the monitoring system is compatible with reported MODC.

Specifications:

If (***Monitor Hourly Modc Status*** == true AND ***Monitor Hourly System Status*** == true AND***Current MHV Mon Sys Record*** is not null)
case (***Current MHV Record.***ModcCode)

01 OR 17: If (***Current MHV Mon Sys Record***. SystemDesignationCode NOT in set {P, PB})
return result A

02: If (***Current MHV Mon Sys Record***. SystemDesignationCode NOT in set {B, RB, DB}
return result B

04: If (***Current MHV Mon Sys Record***. SystemDesignationCode <> "RM")
return result C

22: If (***Current MHV Mon Sys Record***. SystemDesignationCode <> "CI")
return result D

Results:

Result	Response	Severity
A	You reported an MODCCode of [modcCode] in the MHV record for [param], but MonitoringSystemID [ID] is not a primary system.	Critical Error Level 1
B	You reported an MODCCode of 02 in the MHV record for [param], but MonitoringSystemID [ID] is not a backup system.	Critical Error Level 1
C	You reported an MODCCode of 04 in the MHV record for [param], but MonitoringSystemID [ID] is not a reference method system.	Critical Error Level 1
D	You reported an MODCCode of 22 in the MHV record for [param], but MonitoringSystemID [ID] is not a certified inlet system.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-15

Check Name: Check Component in MHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that the component in the MHV record is valid.

Specifications:

Monitor Hourly Component Status = false
Flow Averaging Component List = { empty }

If (***Monitor Hourly MODC Status*** == true*)*

if ***Current MHV Record.*** ComponentID is null

case (***Current MHV Parameter***)

SO2C: *MODC Set* = {01, 02, 03, 04, 16, 17, 18, 19, 20, 21, 22, 53}
NOXC: *MODC Set* = {01, 02, 03, 04, 17, 18, 19, 20, 21, 22, 53}
CO2C, O2D, O2W: *MODC Set* = {01, 02, 03, 04, 17, 18, 20, 21, 53}
FLOW: *MODC Set* = {01, 02, 03, 04, 20, 53}
H2O: *MODC Set* = {01, 02, 03, 04, 21, 53}
CO2CSD, O2CSD: *MODC Set* = { }

If (***Current MHV Parameter*** == "FLOW" and ***Current MHV Record***.MonitoringSystemID is not null)

Flow Averaging Component Records = Active MonitoringSystemComponent record for location where:

1) MonitoringSystemID = ***Current MHV Record.***MonitoringSystemID.
2) ComponentTypeCode = "FLOW".

If (Count of *Flow Averaging Component Records* < 2)

If (***Current MHV Record***.ModcCode in set *MODC Set*)
return result A
Else
return result F
Else
Add each record (limited to 2) in *Flow Averaging Component Records* to ***Flow Averaging Component List***
Monitor Hourly Component Status = true

Otherwise,
If (***Current MHV Record***.ModcCode in set *MODC Set*)
return result A
Else
return result F
else
Current MHV Component Record = find ComponentData record where ComponentData.ComponentID = ***Current MHV Record.*** ComponentID

If *Current MHV Component Record*.ComponentTypeCode <> ***Current MHV Component Type***
return result B

else if ***Current MHV Record.***ModcCode == 17 AND *Current MHV Component Record*. ComponentIdentifier does not begin with "LK"
return result C

else if *Current MHV Component Record*.ComponentIdentifier begins with "LK" AND ***Current MHV Record.***ModcCode in set { 01, 02, 03, 04, 05 }
return result G

else if (***Monitor Hourly System Status*** == true AND ***Current MHV Mon Sys Record*** is not null)

Count Mon Sys Comp Record = count active MonitoringSystemComponent record for location where
MonitoringSystemComponentData .ComponentID = ***Current MHV Record***.ComponentID AND
MonitoringSystemComponentData.MonitoringSystemID = ***Current MHV Record.***MonitoringSystemID

If *Count Mon Sys Comp Record* = 0
return result D
Else
Monitor Hourly Component Status= true
Else
Monitor Hourly Component Status = true

Results:

Result	Response	Severity
A	You did not report a ComponentID in the MHV record for [param].	Critical Error Level 1
B	You reported ComponentID [ID] in the MHV record for [param], but this is not an [comptype] component.	Critical Error Level 1
C	You reported an MODCCode of 17 in the MHV record for [param], which indicates that the component is a like-kind analyzer, but the ComponentID does not begin with LK.	Critical Error Level 1
D	You reported MonitoringSystemID [sys] ComponentID [ID] in the MHV record for [param], but there is no MonitorSystemComponent record for this system and component in your monitoring plan that was active during the hour.	Critical Error Level 1
E	You reported a ComponentID in the MHV record for [param], but this field should be blank whenever missing data substitution is performed.	Non-Critical Error
F	You did not report a ComponentID in the MHV record for [param]. The ComponentId is required when you report for missing data substitution.	Critical Error Level 1
G	You reported a ComponentID in the [param] MHV record that begins with "LK", but did not report an MODCCode of 17. You must report an MODCCode of 17 when a like-kind analyzer is used.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data
3	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data
9	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-16

Check Name: Check Unadjusted Value

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that the UnadjustedValue in the MHV record for SO2C, NOXC, and FLOW is valid.

Specifications:

Monitor Hourly Preadjusted Value Status = false

SO2C: *MODC Set* = {01, 02, 03, 04, 16, 17, 18, 19, 20, 21, 22, 53, 54}
NOXC: *MODC Set* = {01, 02, 03, 04, 17, 18, 19, 20, 21, 22, 53, 54}
FLOW: *MODC Set* = {01, 02, 03, 04, 20, 53, 54}

If (***Current MHV Record.***ModcCode in set *MODC Set*)

If (***Current MHV Record***.UnadjustedHourlyValue is null AND
Current MHV Record.ModcCode not in set {04, 19, 20, 53, 54})
return result A

else if (***Current MHV Record***.UnadjustedHourlyValue < 0.0 AND
Current MHV Record.ModcCode not in set {16, 21})
return result A

else if (***Current MHV Record***.UnadjustedHourlyValue == 2 AND
Current MHV Record.ModcCode == 16)
return result G

else if (***Current MHV Record***.UnadjustedHourlyValue > 2 AND
Current MHV Record.ModcCode == 16)
return result B

else if (***Current MHV Record***.UnadjustedHourlyValue > 0 AND
Current MHV Record.ModcCode == 21)
return result C

else if (***Current MHV Parameter*** in set {SO2C, NOXC} and ***Current MHV Record***.UnadjustedHourlyValue is not rounded to 1 decimal place)
return result F

else if (***Current MHV Parameter*** == "FLOW" and ***Current MHV Record***.UnadjustedHourlyValue is not rounded to the nearest 1000)
return result F

else
Monitor Hourly Preadjusted Value Status = true

if (***Current MHV Max Min Value*** is not null)
if (***Current MHV Record***.UnadjustedHourlyValue > ***Current MHV Max Min Value***)
return result D

else if (***Monitor Hourly Modc Status*** == true)

If (***Current MHV Record***.UnadjustedHourlyValue is not null)
If (***Current MHV Record.***ModcCode == "46")
return result H
Else

return result E

Else

Monitor Hourly Preadjusted Value Status = true

Results:

Result	Response	Severity
A	The UnadjustedHourlyValue reported in the MHV record for [param] is missing or invalid.	Critical Error Level 1
B	You reported an MODCCode of 16 in the MHV record for [param], but the UnadjustedHourlyValue exceeds 2.	Critical Error Level 1
C	You reported an MODCCode of 21 in the MHV record for [param], but the UnadjustedHourlyValue is greater than 0.	Critical Error Level 1
D	Warning: The UnadjustedHourlyValue reported in the MHV record for [param] is in excess of the maximum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update Span and/or Default values. You should investigate the cause of these exceedances and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Informational Message
E	You reported an MODCCode of [modcCode] in the MHV record for [param], so you should not have reported a value for the UnadjustedHourlyValue.	Critical Error Level 1
F	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
G	You reported an MODCCode of 16 in the MHV record for [param], but the UnadjustedHourlyValue is equal to 2. According to Part 75.11(e)(3)(iii) any bias-adjusted hourly average SO2 concentration of less than 2.0 ppm recorded by the SO2 monitoring system shall be adjusted to a default value of 2.0 ppm, for reporting purposes.	Informational Message
H	You reported an MODCCode of [modcCode] in the MHV record for [param], so you should not have reported a value for the UnadjustedHourlyValue.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-17

Check Name: Verify Consistency Between NOx Emission Rate and NOx Concentration

Related Former Checks:

Applicability: CEM Check

Description: This check ensures consistency between NOx emission rate records and NOx Concentration records based on the MODC and reported values.

Specifications:

If (***NOx Conc Needed for NOx Rate Calc*** == true AND ***Monitor Hourly Modc Status*** == true)

if (***Current MHV Record***.MODCCode not in set {01, 02, 03, 04, 17, 18, 19, 20, 21, 22, 46, 53})
if (***NOx Emission Rate MODC*** in set {01, 02, 03, 04, 14, 21, 22, 53, 54})
return result A
else if (***Current MHV Record***.MODCCode == 21 AND ***NOx Emission Rate MODC*** not in set {14, 21})
return result A
else if (***Current MHV Record***.MODCCode == 22 AND ***NOx Emission Rate MODC*** not in set{14, 22})
return result A
else if (***Current MHV Record***.MODCCode == 46 AND ***NOx Emission Rate MODC*** in set {01, 02, 03, 04, 05, 14, 21, 22, 53, 54})
return result B

Results:

Result	Response	Severity
A	You reported an MODCCode of [MODC] in the MHV record for NOXC that is inconsistent with MODCCode of [NOX ER MODC] reported in the DHV record for NOXR.	Critical Error Level 1
B	You reported an MODCCode of [MODC] in the MHV record for NOXC that is inconsistent with MODCCode of [NOX ER MODC] reported in the DHV record for NOXR.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation

Check Code: HOURMHV-18

Check Name: Determine Maximum or Minimum Value for Parameter in MHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check determines the maximum or minimum value for the parameter from the span or default table based on MODC.

Specifications:

Current MHV Max Min Value = null

If (***Current MHV Parameter*** == "H2O")
    If (***H2O Missing Data Approach*** == "MAX")
        Current MHV Default Parameter = "H2OX"
    Else If (***H2O Missing Data Approach*** == "MIN")
        Current MHV Default Parameter = "H2ON"
    else if (***Current MHV Record.***ModcCode == 12)
        return result A

If (***Monitor Hourly Modc Status*** == true AND ***Current MHV Default Parameter*** is not null)

    If (***Current MHV Record.***ModcCode in set {12, 23} AND ***Current MHV Fuel Specific Hour*** = true)

        If ***Current Hourly Op Record***.FuelCode is not null

            Current MHV Missing Data Fuel = ***Current Hourly Op Record***.FuelCode

            Count active MonitoringDefaultData record for location where
                ParameterCode = ***Current MHV Default Parameter***
                FuelCode = ***Current Hourly Op Record***.FuelCode
                DefaultPurposeCode = "MD" // Missing Data
                OperatingCode in set {A,U} // Not Controlled

            if (*count* > 1)
                return result B
            else if (*count* == 0)
                return result C
            else
                Default Record = the single matched record

                if (*Default Record*.DefaultValue > 0)
                    Current MHV Max Min Value = *Default Record*.DefaultValue
                else
                    return result D

    else if (***Current MHV Record.***ModcCode in set {13, 24} AND ***Current MHV Fuel Specific Hour*** == true)

        If ***Current Hourly Op Record***.FuelCode is not null

            Current MHV Missing Data Fuel = ***Current Hourly Op Record***.FuelCode

            Count active MonitoringDefaultData record for location where
                ParameterCode = ***Current MHV Default Parameter***
                FuelCode = ***Current Hourly Op Record***.FuelCode
                DefaultPurposeCode = "MD" // Missing Data
                OperatingCode in == "C" // Controlled

```
                    if (count > 1)
                        return result B
                    else if (count == 0)
                        return result C
                    else
                        Default Record = the single matched record

                        if (Default Record.DefaultValue > 0)
                            Current MHV Max Min Value = Default Record.DefaultValue
                        else
                            return result D

else if (Current MHV Record.ModcCode <> 15)

        If (Current MHV Parameter  in set {H2O, O2W, O2D, O2CSD})

            if (Current MHV Default Parameter is not null)

                Current MHV Missing Data Fuel = "NFS"

                if (Current MHV Parameter in set {O2W, O2D} AND Current MHV Record.ModcCode == 20)
                    Current MHV Default Parameter = "O2X"

                    Count active MonitoringDefaultData record for location where
                        ParameterCode = Current MHV Default Parameter
                        FuelCode = "NFS"
                        DefaultPurposeCode = "DC" // diluent cap

                else

                    Count active MonitoringDefaultData record for location where
                        ParameterCode = Current MHV Default Parameter
                        FuelCode = "NFS"
                        DefaultPurposeCode =  "MD" // missing data

                if (count > 1)
                    return result B

                else if ((Current MHV Parameter  == "O2D" AND O2 Dry Checks Needed for Heat Input == false) OR
                (Current MHV Parameter  == "O2W" AND O2 Wet Checks Needed for Heat Input == false))
                    Current MHV Max Min Value = 0

                else if (count == 0)
                    return result C

                else
                    Default Record = the single matched record

                    if (Default Record.DefaultValue > 0)
                        Current MHV Max Min Value = Default Record.DefaultValue
                    else
                        return result D

        else

            If (Current MHV Component Type == "FLOW")
                Current MHV Span Description = "FLOW"
```

```
        Monitor Span Record Count = Find active MonitoringSpanData records for location where
                MonitoringSpanData .ComponentTypeCode = "FLOW"
else
        Current MHV Span Description = Current MHV Component Type + " with a SpanScale of H"
        Monitor Span Record Count = Find active MonitoringSpanData records for location where
                MonitoringSpanData .ComponentTypeCode = Current MHV Component Type AND
                MonitoringSpanData.SpanScaleCode = "H"

if (Monitor Span Record Count > 1)
        return result E
else if (Monitor Span Record Count = 0)
        return result F
else
        Current Monitor Span Record = the single matched record

        If (Current MHV Record.ModcCode == 19)

                if Current Monitor Span Record.DefaultHighRange > 0)
                        Current MHV Max Min Value = Current Monitor Span Record.DefaultHighRange
                else
                        return result G

        else if ((Current Monitor Span Record.DefaultHighRange is null AND Current MHV Record.ModcCode
        not in set {13, 24}) or Current MHV Record.ModcCode == 12)

                if (Current MHV Record.ModcCode == 20)
                        if (Current MHV Parameter == "FLOW")
                                if Current Monitor Span Record.FlowFullScaleRange > 0)
                                        Current MHV Max Min Value = Current Monitor Span
                                        Record.FlowFullScaleRange * 2
                                else
                                        return result G
                        else
                                if Current Monitor Span Record.FullScaleRange > 0)
                                        Current MHV Max Min Value = Current Monitor Span
                                        Record.FullScaleRange * 2
                                else
                                        return result G

                else
                        if (Current MHV Parameter == "FLOW")
                                if Current Monitor Span Record.MPFValue > 0)
                                        Current MHV Max Min Value = Current Monitor Span
                                        Record.MPFValue
                                else
                                        return result G
                        else
                                if Current Monitor Span Record.MPCValue > 0)
                                        Current MHV Max Min Value = Current Monitor Span
                                        Record.MPCValue
                                else
                                        return result G

        else if (Current MHV Parameter in set {SO2C, NOXC})

                Current MHV Span Description = Current MHV Component Type + " with a SpanScale of L"
                Monitor Span Record Count = Find active MonitoringSpanData records for location where
```

```
                    MonitoringSpanData .ComponentTypeCode = Current MHV Component Type AND
                    MonitoringSpanData.SpanScaleCode = "L"

            if (Monitor Span Record Count > 1)
                    return result E

            else if (Monitor Span Record Count = 0)
                    return result F

            else
                    Current Monitor Span Record = the single matched record

                    if (Current MHV Record.ModcCode == 20)

                            if Current Monitor Span Record.FullScaleRange > 0)
                                    Current MHV Max Min Value = Current Monitor Span
                                    Record.FullScaleRange * 2
                            else
                                    return result G

                    else if (Current MHV Record.ModcCode in set {13, 24})
                            if Current Monitor Span Record.MECValue > 0)
                                    Current MHV Max Min Value = Current Monitor Span
                                    Record.MECValue
                            else
                                    return result G

                    else
                            if Current Monitor Span Record.SpanValue > 0)
                                    Current MHV Max Min Value = Current Monitor Span
                                    Record.SpanValue
                            else
                                    return result G
```

Results:

Result	Response	Severity
A	The missing data default parameter for H2O could not be determined, because you used both Standard and Inverse Part 75 missing data approaches during the hour.	Critical Error Level 2
B	You reported more than one applicable [param] Default record with a FuelCode of [FuelCode] in your monitoring plan for the hour.	Critical Error Level 1
C	You did not report an applicable [param] Default record with a FuelCode of [FuelCode].	Critical Error Level 1
D	The values reported in the applicable [param] Default record with a FuelCode of [FuelCode] are invalid.	Critical Error Level 1
E	You reported more than one active span record for [key] in your monitoring plan for the hour.	Critical Error Level 1
F	You did not report an active span record for [key] in your monitoring plan for the hour.	Critical Error Level 1
G	The values reported in the applicable span record for [key] are invalid.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data
3	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data
9	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-19

Check Name: Check Adjusted Hourly Value in MHV Record

Related Former Checks:

Applicability: CEM Check

Description: This checks ensures that AdjustedHourlyValue is valid and does not conflict with the reported MODC codes.

Specifications:

Monitor Hourly Adjusted Value Status = false

if (***Monitor Hourly Modc Status*** == true AND ***Monitor Hourly Missing Data Status*** == true AND
(***Monitor Hourly Pma Status*** == true OR ***Current MHV Record***.ModcCode NOT in set {06, 07, 08, 09, 10, 11}))

If (***Current MHV Parameter*** <> "NOXC" OR ***NOx Conc Needed for Nox Mass Calc*** == true)

If (Current MHV Parameter == "FLOW")
set *Current MHV Precision* to -3.
else
set *Current MHV Precision* to 1.

case (***Current MHV Record.***ModcCode)

= 21: ***Current MHV Calculated Adjusted Value*** = 0
if (***Current MHV Record***.AdjustedHourlyValue == 0)
Monitor Hourly Adjusted Value Status = true
else
return result A

= 16: ***Current MHV Calculated Adjusted Value*** = 2
if (***Current MHV Record***.AdjustedHourlyValue == 2)
Monitor Hourly Adjusted Value Status = true
else
return result B

= 12 OR = 23:
If (***Current MHV Max Min Value*** is not null)
Current MHV Calculated Adjusted Value = ***Current MHV Max Min Value***
if (***Current MHV Record***.AdjustedHourlyValue == ***Current MHV Max Min Value***)
Monitor Hourly Adjusted Value Status = true
else
return result C

= 13 OR 24:
If (***Current MHV Max Min Value*** is not null)
Current MHV Calculated Adjusted Value = ***Current MHV Max Min Value***
if (***Current MHV Record***.AdjustedHourlyValue == ***Current MHV Max Min Value***)
Monitor Hourly Adjusted Value Status = true
else
return result D

= 06: If (***Current MHV HBHA Value*** is not null)

Current MHV Calculated Adjusted Value = ***Current MHV HBHA Value***

If (***Current MHV Record***.AdjustedHourlyValue >= 0)
if (***Current MHV Record***.AdjustedHourlyValue == ***Current MHV Calculated Adjusted Value***)
Monitor Hourly Adjusted Value Status = true

else
return result G
else
return result H

else
If (***Current MHV Record***.AdjustedHourlyValue >= 0)
If (***Current MHV Record***.AdjustedHourlyValue is not rounded to *Current MHV Precision*)
return result L
else
Current MHV Calculated Adjusted Value = ***Current MHV Record***.AdjustedHourlyValue
Monitor Hourly Adjusted Value Status = true

If (***Current MHV Max Min Value*** is not null)
if (***Current MHV Record***.AdjustedHourlyValue > ***Current MHV Max Min Value***)
If (***Current MHV Parameter*** == "SO2C" and ***Current MHV Record***.AdjustedHourlyValue > ***Current MHV Max Min Value*** * 2)
return result O
Otherwise,
return result K

Else
return result H

= 08 OR 09:

If (***Current MHV Record***.AdjustedHourlyValue >= 0)

If (***Current MHV HBHA Value*** is not null AND ***Current MHV HBHA Value*** > ***Current MHV Record***.AdjustedHourlyValue AND (***Unit is Load Based*** == true or ***Current MHV Parameter*** <> "NOXC")
Current MHV Calculated Adjusted Value = ***Current MHV HBHA Value***
return result I
else
If (***Current MHV Record***.AdjustedHourlyValue is not rounded to *Current MHV Precision*)
return result L
else
Current MHV Calculated Adjusted Value = ***Current MHV Record***.AdjustedHourlyValue
Monitor Hourly Adjusted Value Status = true

If (***Current MHV Max Min Value*** is not null)
if (***Current MHV Record***.AdjustedHourlyValue > ***Current MHV Max Min Value***)
If (***Current MHV Parameter*** == "SO2C" and ***Current MHV Record***.AdjustedHourlyValue > ***Current MHV Max Min Value*** * 2)
return result O
Otherwise,
return result K

Else
return result H

= 04, 05, 07, 10, 11, 15, 53, 54, OR 55:

If (***Current MHV Record***.AdjustedHourlyValue >= 0)
If (***Current MHV Record***.AdjustedHourlyValue is not rounded to *Current MHV Precision*)
return result L
else
Current MHV Calculated Adjusted Value = ***Current MHV Record***.AdjustedHourlyValue
Monitor Hourly Adjusted Value Status = true

If (***Current MHV Max Min Value*** is not null)
if (***Current MHV Record***.AdjustedHourlyValue > ***Current MHV Max Min Value***)
If (***Current MHV Parameter*** == "SO2C" and ***Current MHV Record***.AdjustedHourlyValue > ***Current MHV Max Min Value*** * 2)
If the Current MHV Record.ModcCode == 10
return result P
Otherwise,
return result O
Otherwise,
return result K

Else
return result H

= All Other Codes:

If (***Current MHV Record***.AdjustedHourlyValue >= 0)
If (Current MHV Record.ModcCode in set {19, 20} AND ***Current MHV Record***.UnadjustedHourlyValue is null)
If (***Current MHV Max Min Value*** is not null)

if (***Current MHV Record***.AdjustedHourlyValue == ***Current MHV Max Min Value***)
Current MHV Calculated Adjusted Value = ***Current MHV Record***.AdjustedHourlyValue
Monitor Hourly Adjusted Value Status = true
else
if (***Current MHV Record.***ModcCode == "19")
return result M
else
return result N

else if (***Current MHV Record***.AdjustedHourlyValue is not rounded to *Current MHV Precision*)
return result L
else
Monitor Hourly Adjusted Value Status = true
Else
return result H

else

If (***Current MHV Record***.AdjustedHourlyValue is not null)
return result J

Results:

Result	Response	Severity
A	You reported an MODCCode of 21 in the MHV record for [param], but the AdjustedHourlyValue does not equal 0.	Critical Error Level 1
B	You reported an MODCCode of 16 in the MHV record for [param], but the AdjustedHourlyValue does not equal 2.	Critical Error Level 1
C	You reported an MODCCode of [modcCode] in the MHV record for [param], but the AdjustedHourlyValue does not equal the maximum potential value reported in the [comptype] span or fuel-specific default record in your monitoring plan.	Critical Error Level 1
D	You reported an MODCCode of 13 or 24 in the MHV record for [param], but the AdjustedHourlyValue does not equal the maximum expected value reported in the [comptype] span or fuel-specific default record in your monitoring plan.	Critical Error Level 1
G	You reported an MODCCode of 06 in the MHV record for [param], but the AdjustedHourlyValue does not equal average of measured hour before and measured hour after.	Critical Error Level 1
H	The AdjustedHourlyValue reported in the MHV record for [param] is invalid.	Critical Error Level 1
I	You reported an MODCCode of [MODCCode] in the MHV record for [param], but you reported an AdjustedHourlyValue that is less than the average of the measured hour before and measured hour after.	Critical Error Level 1
J	You reported an AdjustedHourlyValue in the MHV record for NOXC. This field should be blank when the NOX concentration is used to calculate the NOX emission rate as part of a NOX system.	Critical Error Level 1
K	Warning: The AdjustedHourlyValue reported in the MHV record for [param] is in excess of the maximum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update Span and/or Default values. You should investigate the cause of these exceedances and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Informational Message
L	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
M	You reported an MODCCode of 19 in the MHV record for [param], but the AdjustedHourlyValue does not equal the DefaultHighRange reported in the [comptype] span record in your monitoring plan.	Critical Error Level 1
N	You reported an MODCCode of 20 in the MHV record for [param], but the AdjustedHourlyValue does not equal 200 percent of the FullScaleRange reported in the [comptype] span record in your monitoring plan.	Critical Error Level 1
O	The AdjustedHourlyValue reported in the MHV record for [param] is in excess of the maximum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update Span and/or Default values. You should investigate the cause of these exceedances and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Critical Error Level 2
P	The AdjustedHourlyValue reported in the MHV record for [param] is in excess of the maximum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update Span and/or Default values. You should investigate the cause of these exceedances and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-20

Check Name: Check Unadjusted Hourly Value in MHV Record

Related Former Checks:

Applicability: CEM Check

Description: This checks ensures that UnadjustedHourlyValue is valid and does not conflict with the reported MODC codes.

Specifications:

Monitor Hourly Unadjusted Value Status = false

if (***Monitor Hourly Modc Status*** == true AND ***Monitor Hourly Missing Data Status*** == true AND
(***Monitor Hourly Pma Status*** == true OR ***Current MHV Record***.ModcCode NOT in set {06, 07, 08, 09, 10, 11}))

Calculated Unadjusted Value = null

case (***Current MHV Record.***ModcCode)

= 21: *Calculated Unadjusted Value* = 0
if (***Current MHV Record***.UnadjustedHourlyValue == 0)
If (***Current MHV Parameter*** == "CO2C" and ***Current Hourly Op Record***.LoadRange is greater than 1,
return result L
else
Monitor Hourly Unadjusted Value Status = true
else
return result A

= 12 OR = 23:
If (***Current MHV Max Min Value*** is not null)
Calculated Unadjusted Value = ***Current MHV Max Min Value***
if (***Current MHV Record***.UnadjustedHourlyValue == ***Current MHV Max Min Value***)
Monitor Hourly Unadjusted Value Status = true
else
return result B

= 20: If (***Current MHV Record***.UnadjustedHourlyValue >= 0)
If (***Current MHV Max Min Value*** is not null)
If (***Current MHV Parameter*** begins with "O2" AND ***Current MHV Record***.UnadjustedHourlyValue > 20.9)
Calculated Unadjusted Value = ***Current MHV Max Min Value***
return result K

else if ***Current MHV Parameter*** == "CO2C" AND ***Current MHV Record***.UnadjustedHourlyValue > ***Current MHV Max Min Value***)
Calculated Unadjusted Value = ***Current MHV Max Min Value***
return result C

elseif (***Current MHV Record***.UnadjustedHourlyValue is not rounded to one decimal place)
return result I
else
Calculated Unadjusted Value = ***Current MHV Record***.UnadjustedHourlyValue
Monitor Hourly Unadjusted Value Status = true
else
return result E

= 06: If (***Current MHV HBHA Value*** is not null)

Calculated Unadjusted Value = ***Current MHV HBHA Value***

If (***Current MHV Record***.UnadjustedHourlyValue >= 0)
if (***Current MHV Record***.UnadjustedHourlyValue == *Calculated Unadjusted Value*)
Monitor Hourly Unadjusted Value Status = true
else
return result D
else
return result E

else
If (***Current MHV Record***.UnadjustedHourlyValue >= 0)
If (***Current MHV Record***.UnadjustedHourlyValue is not rounded to one decimal place)
return result I
else
Calculated Unadjusted Value = ***Current MHV Record***.UnadjustedHourlyValue
Monitor Hourly Unadjusted Value Status = true

if (***Current MHV Max Min Value*** is not null)
If ((***Current MHV Parameter*** == "H2O" AND ***H2O Missing Data Approach*** == "MIN")
OR ***Current MHV Parameter*** begins with "O2")
if (***Current MHV Record***.UnadjustedHourlyValue < ***Current MHV Max Min Value***)
return result H
else
if (***Current MHV Record***.UnadjustedHourlyValue > ***Current MHV Max Min Value***)
return result F
Else
return result E

= 08 OR = 09:

If (***Current MHV Record***.UnadjustedHourlyValue >= 0)

If (***Current MHV HBHA Value*** is not null AND ((***Current MHV Parameter*** == "H2O" AND ***H2O Missing Data Approach*** == "MIN") OR ***Current MHV Parameter*** begins with "O2") AND ***Current MHV HBHA Value*** < ***Current MHV Record***.UnadjustedHourlyValue)
Calculated Unadjusted Value = ***Current MHV HBHA Value***
return result J

else if (***Current MHV HBHA Value*** is not null AND ((***Current MHV Parameter*** == "H2O" AND ***H2O Missing Data Approach*** == "MAX") OR ***Current MHV Parameter*** does not begin with "O2" or "H2O") AND ***Current MHV HBHA Value*** > ***Current MHV Record***.UnadjustedHourlyValue)
Calculated Unadjusted Value = ***Current MHV HBHA Value***
return result G
else
if (***Current MHV Record***.UnadjustedHourlyValue is not rounded to one decimal place)
return result I
else
Calculated Unadjusted Value = ***Current MHV Record***.UnadjustedHourlyValue
Monitor Hourly Unadjusted Value Status = true

if (***Current MHV Max Min Value*** is not null)
If ((***Current MHV Parameter*** == "H2O" AND ***H2O Missing Data Approach*** == "MIN")
OR ***Current MHV Parameter*** begins with "O2")
if (***Current MHV Record***.UnadjustedHourlyValue < ***Current MHV Max Min Value***)

return result H

else

if (***Current MHV Record***.UnadjustedHourlyValue > ***Current MHV Max Min Value***)

return result F

Else

return result E

= 46:

If (***Current MHV Record***.UnadjustedHourlyValue != null)

return result M

= All Other Codes:

If (***Current MHV Record***.UnadjustedHourlyValue >= 0)

If (***Current MHV Parameter*** in set {"H2O", "CO2C", "O2D", "O2W", "CO2CSD", "O2CSD"} AND ***Current MHV Record***.UnadjustedHourlyValue > 100)

return result E

else if (***Current MHV Record***.UnadjustedHourlyValue is not rounded to one decimal place)

return result I

else if (***Current MHV Record***.UnadjustedHourlyValue == 0 AND ***Current MHV Parameter*** == "CO2C" and ***Current Hourly Op Record***.LoadRange is greater than 1,

return result L

else

Calculated Unadjusted Value = ***Current MHV Record***.UnadjustedHourlyValue

Monitor Hourly Unadjusted Value Status = true

if (***Current MHV Max Min Value*** is not null)

If ((***Current MHV Parameter*** == "H2O" AND ***H2O Missing Data Approach*** == "MIN") OR ***Current MHV Parameter*** begins with "O2")

if (***Current MHV Record***.UnadjustedHourlyValue < ***Current MHV Max Min Value***)

return result H

else

if (***Current MHV Record***.UnadjustedHourlyValue > ***Current MHV Max Min Value***)

return result F

Else

return result E

Whether or not there is a result returned:
If (*Calculated Unadjusted Value* is not null)

case (***Current MHV Parameter***)

CO2C: ***CO2C MHV Calculated Adjusted Value*** = *Calculated Unadjusted Value*
O2W: ***O2 Wet Calculated Adjusted Value*** = *Calculated Unadjusted Value*
O2D: ***O2 Dry Calculated Adjusted Value*** = *Calculated Unadjusted Value*
H2O: ***H2O MHVCalculated Adjusted Value*** = *Calculated Unadjusted Value*
CO2CSD: ***CO2C SD Calculated Adjusted Value*** = *Calculated Unadjusted Value*
O2CSD: ***O2C SD Calculated Adjusted Value*** = *Calculated Unadjusted Value*

Results:

Result	Response	Severity
A	You reported an MODCCode of 21 in the MHV record for [param], but the UnadjustedHourlyValue does not equal 0.	Critical Error Level 1
B	You reported an MODCCode of [modc] in the MHV record for [param], but the UnadjustedHourlyValue does not equal the maximum potential value reported in the [comptype] span or default record in your monitoring plan.	Critical Error Level 1
C	You reported an MODCCode of 20 in the MHV record for [param], but the UnadjustedHourlyValue does not equal 200 percent of the FullScaleRange reported in the CO2 span record in your monitoring plan.	Critical Error Level 1
D	You reported an MODCCode of 06 in the MHV record for [param], but the UnadjustedHourlyValue does not equal average of measured hour before and measured hour after.	Critical Error Level 1
E	The UnadjustedHourlyValue reported in the MHV record for [param] is missing or invalid.	Critical Error Level 1
F	Warning: The UnadjustedHourlyValue reported in the MHV record for [param] is in excess of the maximum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update Span and/or Default values. You should investigate the cause of these exceedances and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Informational Message
G	You reported an MODCCode of [MODC] in the MHV record for [param], but you reported an UnadjustedHourlyValue that is less than the average of the measured hour before and measured hour after.	Critical Error Level 1
H	Warning: The UnadjustedHourlyValue reported in the MHV record for [param] is lower than the minimum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these minimum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update Default values. You should investigate the cause of these low values and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Informational Message
I	You reported [fieldname] in the [type] record for [param] that is not rounded to the appropriate precision for that parameter.	Critical Error Level 1
J	You reported an MODCCode of [MODC] in the MHV record for [param], but you reported an UnadjustedHourlyValue that is greater than the average of the measured hour before and measured hour after.	Critical Error Level 1
K	You reported an MODCCode of 20 in the MHV record for [param], but the UnadjustedHourlyValue does not equal the default value reported in the O2X default record in your monitoring plan.	Critical Error Level 1
L	You have reported an UnadjustedHourlyValue of 0 in the MHV record for [param], but the LoadRange is greater than 1. Emissions for [param] should be greater than 0 when the unit (or stack) is operating at this load level.	Critical Error Level 1
M	You reported an UnadjustedHourlyValue with a MODCCode of [MODC] in the MHV record for [param]. MODCCode [MODC] requires a null UnadjustedHourlyValue.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data
3	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data

Check Code: HOURMHV-21

Check Name: Determine BAF Value for Monitoring System in MHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check retrieves and sets as an output parameter the Bias Adjustment factor for the Monitoring System

Specifications:

Current SO2 System BAF = null
Current NOXC System BAF = null
Current FLOW System BAF = null

Continue = true
If (***Current MHV Parameter*** == "NOXC")
    If (***NOx Conc Needed for Nox Mass Calc*** == false)
        Continue == false

If (*Continue* == true AND ***Monitor Hourly System Status*** == true AND ***Monitor Hourly Preadjusted Value Status*** == true AND (***Current MHV Record.***ModcCode in set {01, 02, 03, 17, 18, 22, 53} OR (***Current MHV Record.***ModcCode in set {19, 20} AND ***Current MHV Record.***UnadjustedHourlyValue is not null AND ***Current MHV Max Min Value*** is not null)))

    If (***RATA Status BAF*** is not null)

        case (***Current MHV Parameter***)
            SO2C: ***Current SO2 System BAF*** = ***RATA Status BAF***
            NOXC: ***Current NOXC System BAF*** = ***RATA Status BAF***
            FLOW: ***Current FLOW System BAF*** = ***RATA Status BAF***

    else
        return result A

Results:

Result	Response	Severity
A	The BAF for [ParamCode] MonitoringSystemID [ID] cannot be determined, because the prior RATA had critical errors or because of a RATA Status error listed on this report.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- NOXC RATA Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- SO2 RATA Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- Stack Flow RATA Status Evaluation

Check Code: HOURMHV-22

Check Name: Calculate Bias Adjusted Value in MHV Record

Related Former Checks:

Applicability: CEM Check

Description: This check ensures that the reported Unadjusted Hourly Value multiplied by the BAF results in the reported Adjusted Hourly Value

Specifications:

case (***Current MHV Parameter***)
SO2C: *Current BAF* = ***Current SO2 System BAF***
NOXC: *Current BAF* = ***Current NOXC System BAF***
FLOW: *Current BAF* = ***Current FLOW System BAF***

if (*Current BAF* is not null)
If (***Current MHV Parameter*** == "FLOW")
Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = ***Current MHV Parameter*** AND
UOM = "SCFH"

Calculated Adjusted Value = ***Current MHV Record***.UnadjustedHourlyValue * *Current BAF*, and the result to the nearest 1000.
else
Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = ***Current MHV Parameter*** AND
UOM = "PPM"

Calculated Adjusted Value = ***Current MHV Record***.UnadjustedHourlyValue * *Current BAF*, and the result to one decimal place (0.1).

If (***Current MHV Record***.ModcCode in set {19, 20} AND *Calculated Adjusted Value* **> *Current MHV Max Min Value***)

case (***Current MHV Parameter***)
SO2C: ***SO2C Calculated Adjusted Value*** = ***Current MHV Max Min Value***
NOXC: ***NOXC Calculated Adjusted Value*** = ***Current MHV Max Min Value***
FLOW: ***FLOW Calculated Adjusted Value*** = ***Current MHV Max Min Value***

if (***Monitor Hourly Adjusted Value Status*** == true)
if (***Current MHV Record***.AdjustedHourlyValue <> ***Current MHV Max Min Value***)
If (***Current MHV Record***.ModcCode == 20)
return result A
else
return result C
else

case (***Current MHV Parameter***)
SO2C: ***SO2C Calculated Adjusted Value*** = *Calculated Adjusted Value*
NOXC: ***NOXC Calculated Adjusted Value*** = *Calculated Adjusted Value*
FLOW: ***FLOW Calculated Adjusted Value*** = *Calculated Adjusted Value*

if (***Monitor Hourly Adjusted Value Status*** == true)
if ABS(*Calculated Adjusted Value* - ***Current MHV Record***.AdjustedHourlyValue) > *Tolerance*
return result B
else
case (***Current MHV Parameter***)
SO2C: ***SO2C Calculated Adjusted Value*** = ***Current MHV Calculated Adjusted Value***
NOXC: ***NOXC Calculated Adjusted Value*** = ***Current MHV Calculated Adjusted Value***

FLOW: ***FLOW Calculated Adjusted Value* = *Current MHV Calculated Adjusted Value***

Results:

Result	Response	Severity
A	You reported an MODCCode of 20 in the MHV record for [param], but the AdjustedHourlyValue does not equal 200 percent of the FullScaleRange reported in the [comptype] span record in your monitoring plan.	Critical Error Level 1
B	The AdjustedHourlyValue reported in the MHV record for [param] is not equal to the UnadjustedHourlyValue times the BAF for the [systype] MonitoringSystemID [ID].	Critical Error Level 1
C	You reported an MODCCode of 19 in the MHV record for [param], but the AdjustedHourlyValue does not equal the DefaultHighRange reported in the [comptype] span record in your monitoring plan.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- NOXC RATA Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- SO2 RATA Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- Stack Flow RATA Status Evaluation

Check Code: HOURMHV-23

Check Name: Initialize CO2C Hourly Monitor for Substitute Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameter and output parameter for subsequent monitor hourly checks for CO2C for substitute data when two CO2C are submitted for the hour.

Specifications:

Current MHV Parameter = "CO2CSD"
CO2C SD Calculated Adjusted Value = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data

Check Code: HOURMHV-24

Check Name: Initialize O2C Hourly Monitor for Substitute Data

Related Former Checks:

Applicability: CEM Check

Description: This check sets generic parameter and output parameter for subsequent monitor hourly checks for O2C for substitute data when two O2C with the same moisture basis are submitted for the hour.

Specifications:

Current MHV Parameter = "O2CSD"
O2C SD Calculated Adjusted Value = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data

Check Code: HOURMHV-26

Check Name: Determine if MHV Record Needs QA Status Evaluation

Related Former Checks:

Applicability: CEM Check

Description: Determine if MHV Record Needs QA Status Evaluation

Specifications:

Set ***LinearityStatusRequired*** = false.
Set ***Current Linearity Status*** = null.
Set ***RATAStatusRequired*** = false.
Set ***Current RATA Status*** = null.
Set ***RATA Status BAF*** = null.
Set ***DailyCalStatusRequired*** = false
Set ***Current Daily Cal Status*** = null.
Set ***F2L Status Required*** = false.
Set ***Daily Int Status Required*** = false
Set ***Leak Status Required*** = false

Set ***QaStatusComponentId*** = ***CurrentMHVRecord***.ComponentId
Set ***QaStatusComponentIdentifier*** = ***CurrentMHVRecord***.ComponentIdentifier
Set ***QaStatusComponentTypeCode*** = ***CurrentMHVRecord***.ComponentTypeCode
Set ***QaStatusSystemDesignationCode*** = ***CurrentMHVRecord***.SystemDesignationCode
Set ***QaStatusSystemId*** = ***CurrentMHVRecord***.SystemId
Set ***QaStatusSystemIdentifier*** = ***CurrentMHVRecord***.SystemIdentifier
Set ***QaStatusSystemTypeCode*** = ***CurrentMHVRecord***.SystemTypeCode

If (***PrimaryBypassActiveForHour*** is false)

Set ***QaStatusPrimaryOrPrimaryBypassSystemId*** = null.

Else If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord*** is not null)

Set ***QaStatusPrimaryOrPrimaryBypassSystemId*** = ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.SystemId.

Else If (***CurrentNoxRateDerivedHourlyRecord*** is not null)

Count records in ***MonitorSystemComponentRecordsByHourLocation*** where:

1) ComponentId equals ***QaStatusComponentId***.
2) SystemId equals ***CurrentNoxRateDerivedHourlyRecord***.SystemId.

If (count is greater than 0)
Set ***QaStatusPrimaryOrPrimaryBypassSystemId*** = ***CurrentNoxRateDerivedHourlyRecord***.SystemId.
Else
Set ***QaStatusPrimaryOrPrimaryBypassSystemId*** = null.

Else
Set ***QaStatusPrimaryOrPrimaryBypassSystemId*** = null.

Locate the record in ***MonitorSystemComponentRecordsByHourLocation*** with the earliest BeginDate/BeginHour and ComponentId equal to ***QaStatusComponentId***.

If found,
Set ***QaStatusComponentBeginDate*** = ***MonitorSystemComponentRecordsByHourLocation***.BeginDate

Set ***QaStatusComponentBeginDatehour*** = ***MonitorSystemComponentRecordsByHourLocation***.BeginDatehour
Else
Set ***QaStatusComponentBeginDate*** = null.
Set ***QaStatusComponentBeginDatehour*** = null.

if (***Monitor Hourly Modc Status*** == true AND
(***CurrentMHVRecord***.ModcCode in set {01, 02, 03, 17, 18, 21, 22, 47, 53} OR
(***CurrentMHVRecord***.ModcCode in set {19, 20} AND ***CurrentMHVRecord***.UnadjustedHourlyValue is not null AND
Current MHV Max Min Value is not null)))

if (***MonitorHourlyComponentStatus*** = true AND ***CurrentMHVRecord***.ComponentID is not null AND ***CurrentMHVParameter*** in set {SO2C, NOXC, CO2C, O2D, O2W})

Set ***LinearityStatusRequired*** = true.
Set ***DailyCalStatusRequired*** = true.

if (***MonitorHourlyComponentStatus*** = true AND ***CurrentMHVRecord***.ComponentID is not null AND ***CurrentMHVParameter*** in set {FLOW})

Set ***DailyCalStatusRequired*** = true.
Set ***Daily Int Status Required*** = true.

If ***CurrentMHVRecord***.SampleAcquistionMethodCd = "DP"
Set ***Leak Status Required*** = true.

if (***MonitorHourlySystemStatus*** = true AND ***CurrentMHVRecord***.MonitoringSystemID is not null AND ***CurrentMHVRecord***.SystemTypeCode is in {SO2, NOXC, FLOW, H2OM})

Set ***RATAStatusRequired*** = true.
Set ***CurrentHourlyRecordforRATAStatus*** = ***Current MHV Record***.

if ***CurrentMHVRecord***.SystemTypeCode is equal to 'FLOW',

Set ***F2LStatus Required*** = true.

else if ((***CO2 Conc Checks Needed for Heat Input*** == true AND ***CurrentMHVParameter*** == "CO2C") OR (***O2 Wet Checks Needed for Heat Input*** == true AND ***CurrentMHVParameter*** == "O2W") OR (***O2 Dry Checks Needed for Heat Input*** == true AND ***CurrentMHVParameter*** == "O2D"))

Set ***CO2RATARequired*** = true.

if (***RATAStatus Required*** == false AND ***Current MHV Parameter*** in {SO2C, NOXC, FLOW})

case (***Current MHV Parameter***)
SO2C: ***SO2C Calculated Adjusted Value*** = ***Current MHV Calculated Adjusted Value***
NOXC: ***NOXC Calculated Adjusted Value*** = ***Current MHV Calculated Adjusted Value***
FLOW: ***FLOW Calculated Adjusted Value*** = ***Current MHV Calculated Adjusted Value***

if (***LinearityStatusRequired*** == true OR ***DailyCalStatusRequired*** == true)

Set ***DualRangeStatus*** = false.
Set ***CurrentAnalyzerRangeUsed*** = null.
Set ***ApplicableSystemIDs*** = null.
Set ***HighRangeComponentID*** = null.
Set ***LowRangeComponentID*** = null.

if (***CurrentMHVParameter*** == "FLOW")

Set ***ApplicableComponentID*** = ***Current MHV Record***.ComponentID

else

Set ***ApplicableComponentID*** = null.

Locate a record in ***AnalyzerRangeRecordsByHourLocation*** for the hour and location where the ComponentID is equal to the ***CurrentMHVRecord***.ComponentID.

if (***AnalyzerRangeRecordsByHourLocation*** is not found OR if more than one ***AnalyzerRangeRecordsByHourLocation*** is found)

set ***Linearity Status Required*** == false
set ***DailyCalStatusRequired*** = false
return result A

else

if (***AnalyzerRangeRecordsByHourLocation***.DualRangeIndicator = 1)

Set ***DualRangeStatus*** = true.

if (Analyzer Range Record.AnalyzerRangeCode = "A")

Locate the record in ***MonitorSpanRecordsByHourLocation*** for the hour and location where the ComponentTypeCode is equal to the ***CurrentMHVRecord***.ComponentTypeCode and the SpanScaleCode is equal to "L".

if (***MonitorSpanRecordsByHourLocation*** is not found OR if more than one ***MonitorSpanRecordsByHourLocation*** is found OR if the ***MonitorSpanRecordsByHourLocation***.ScaleTransitionPoint is null or <= 0)

set ***Linearity Status Required*** == false
set ***DailyCalStatusRequired*** = false
return result B

else if (***MonitorSpanRecordsByHourLocation*** is found AND ***CurrentMHVRecord***.UnadjustedHourlyValue > ***MonitorSpanRecordsByHourLocation***.ScaleTransitionPoint AND ***CurrentMHVRecord***.ModcCode <> "18")

Set ***CurrentAnalyzerRangeUsed*** = "H".
Set ***HighRangeComponentID*** = ***CurrentMHVRecord.***Component ID.
Set ***LowRangeComponentID*** = ***CurrentMHVRecord.***Component ID.

else

Set ***CurrentAnalyzerRangeUsed*** = "L".
Set ***HighRangeComponentID*** = ***CurrentMHVRecord.***Component ID.
Set ***LowRangeComponentID*** = ***CurrentMHVRecord.***Component ID.

else

Set ***CurrentAnalyzerRangeUsed*** = ***AnalyzerRangeRecordsByHourLocation***.AnalyzerRangeCode.

if (***AnalyzerRangeRecordsByHourLocation***.AnalyzerRangeCode = "H")

Locate a record in ***AnalyzerRangeRecordsByHourLocation*** for the hour and location where the ComponentTypeCode is equal to the ***CurrentMHVRecord***.ComponentTypeCode and the AnalyzerRangeCode is equal to "L" AND the ComponentSerialNumber is equal to the ***CurrentMHVRecord***.ComponentSerialNumber (removing the phrases "HIGH", "HI", "LOW", and "LO").

if (***AnalyzerRangeRecordsByHourLocation*** is not found OR if more than one ***AnalyzerRangeRecordsByHourLocation*** is found)

set ***Linearity Status Required*** == false
set ***DailyCalStatusRequired*** = false
return result C

else If (***AnalyzerRangeRecordsByHourLocation*** is found)

Set ***HighRangeComponentID*** = ***CurrentMHVRecord.***Component ID.
Set ***LowRangeComponentID*** = ***AnalyzerRangeRecordsByHourLocation.***Component ID.

else if (***CurrentMHVRecord***.AnalyzerRangeCode = "L")

Locate a record in ***AnalyzerRangeRecordsByHourLocation*** for the ***CurrentMHVRecord.***Hour where the ComponentTypeCode is equal to the ***CurrentMHVRecord***.ComponentTypeCode and the AnalyzerRangeCode is equal to "H" AND the ComponentSerialNumber is equal to the ***CurrentMHVRecord***.ComponentSerialNumber (removing the phrases "HIGH", "HI", "LOW", and "LO").

if (***AnalyzerRangeRecordsByHourLocation*** is not found OR if more than one ***AnalyzerRangeRecordsByHourLocation*** is found)

set ***Linearity Status Required*** == false
set ***DailyCalStatusRequired*** = false
return result C

else If (***AnalyzerRangeRecordsByHourLocation*** is found)

Set ***LowRangeComponentID*** = ***CurrentMHVRecord.***Component ID.
Set ***HighRangeComponentID*** = ***AnalyzerRangeRecordsByHourLocation.***Component ID.

else

Set ***CurrentAnalyzerRangeUsed*** = ***AnalyzerRangeRecordsByHourLocation.***AnalyzerRangeCode.

if (***CurrentAnalyzerRangeUsed*** = "H")
Set ***HighRangeComponentID*** = ***CurrentMHVRecord.***Component ID.

else
Set ***LowRangeComponentID*** = ***CurrentMHVRecord.***Component ID.

if (***CurrentAnalyzerRangeUsed*** = "H")
Set ***ApplicableComponentID*** = ***HighRangeComponentID***.

else

Set ***ApplicableComponentID*** = ***LowRangeComponentID***.

For each record in ***MonitorSystemComponentRecordsByHourLocation*** where the ComponentID is equal to the ***ApplicableComponentID***

Append ***MonitorSystemComponentRecordsByHourLocation***.SystemID to ***ApplicableSystemIDs***.

if (***MonitorSystemComponentRecordsByHourLocation*** is not found)

set ***Linearity Status Required*** == false
set ***DailyCalStatusRequired*** = false
return result D

Results:

Result	Response	Severity
A	You did not report one (and only one) valid Analyzer Range record in your monitoring plan for ComponentID [COMPID] for this hour. The QA Status of the linearity and/or daily calibration tests for this component will not be evaluated.	Critical Error Level 1
B	You reported that ComponentID [COMPID] is a dual-range analyzer, but you did not report one (and only one) active low-scale [COMPTYPE] span record with a valid ScaleTransitionPoint in your monitoring plan for this hour. The QA Status of the linearity and/or daily calibration tests for this component will not be evaluated.	Critical Error Level 1
C	You reported that ComponentID [COMPID] is a dual-range analyzer, but the program could not identify the alternate range component in your monitoring plan. The QA Status of the linearity and/or daily calibration tests for this component will not be evaluated.	Critical Error Level 1
D	You did not report any System Component records for ComponentID [compid] in your monitoring plan for the hour. The QA Status of the linearity and/or daily calibration tests for this component will not be evaluated.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ NOXR Unused P-PB Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-27

Check Name: Determine MHV Measure Code

Related Former Checks:

Applicability: CEM Check

Description:

Specifications:

If (***Current MHV Parameter*** == "CO2CSD")
Set ***Monitor Measure Code Array*** for "CO2C" to "SUB"

else If (***Current MHV Parameter*** == "O2CSD")
Set ***Monitor Measure Code Array*** for "O2D" to "SUB"
Set ***Monitor Measure Code Array*** for "O2W" to "SUB"

else if (***Current MHV Parameter*** in set {SO2C, NOXC, CO2C, O2D, O2W, FLOW, H2O} AND ***Monitor Measure Code Array*** for the ***Current MHV Parameter*** is null)
If (***Current MHV Record.***ModcCode in set {01, 02, 03, 04, 05, 16, 17, 19, 20, 21, 22, 53, 54})
Set ***Monitor Measure Code Array*** for the ***Current MHV Parameter*** to "MEASURE"
else if (***Current MHV Record.***ModcCode in set {06, 07, 08, 09, 10, 11, 12, 13, 15, 23, 24, 55})
Set ***Monitor Measure Code Array*** for the ***Current MHV Parameter*** to "SUB"
else if (***Current MHV Record.***ModcCode == "18"
Set ***Monitor Measure Code Array*** for the ***Current MHV Parameter*** to "MEASSUB"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data
3	Process/Category:	Emissions Data Evaluation Report ------ FLOW Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ H2O Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ NOx Concentration Monitor Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data
9	Process/Category:	Emissions Data Evaluation Report ------ SO2 Monitor Hourly Evaluation

Check Code: HOURMHV-28

Check Name: Check Maximum Concentration Percentage Threshold

Related Former Checks:

Applicability: General Check

Description: Checks the percentage reported for CO2C and O2C to ensure they do not exceed 16% and 22% respectively.

Specifications:

When ***CurrentMhvParameter*** equals

"CO2C" or "CO2CSD" then,

If ***CurrentMhvRecord***.ModcCode is equal to "01" or "02", AND ***CurrentMhvRecord***.UnadjustedHourlyValue is greater than 16%,
return result A.

"O2D", "O2W" or "O2CSD" then,

If ***CurrentMhvRecord***.ModcCode is equal to "01" or "02", AND ***CurrentMhvRecord***.UnadjustedHourlyValue is greater than 22%,
return result B.

Results:

Result	Response	Severity
A	The CO2 concentration cannot exceed 16%.	Informational Message
B	The O2 concentration cannot exceed 22%.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ CO2 Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ CO2C Monitor Hourly Evaluation for Substitute Data
3	Process/Category:	Emissions Data Evaluation Report ------ O2 Dry Monitor Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ O2 Wet Monitor Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ O2C Monitor Hourly Evaluation for Substitute Data

Check Code: HOURMHV-29

Check Name: NOxR Primary/Primary Bypass: Initialization by Parameter Code

Related Former Checks:

Applicability: CEM Check

Description: Initializes check parameters based on the ParameterCode of the NOxR Primary/Primary Bypass MHV record.

Specifications:

Set ***CurrentMhvRecord*** to ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.

Set ***CurrentMhvComponentType*** to null.
Set ***CurrentMhvDefaultParame3ter*** to null.
Set ***CurrentMhvParameter*** to null.
Set ***CurrentMhvParameterDescription*** to null.
Set ***CurrentMhvParameterStatus*** to false.

Set ***CompleteMhvRecordNeeded*** to false.
Set ***CurrentMhvFuelSpecificHour*** to false.
Set ***CurrentMhvHbHaValue*** to null.
Set ***CurrentMhvSystemType*** to null.
Set ***MonitorHourlyModcStatus*** to true.

When ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.ParameterCd equals

"CO2C":

If (***Co2DiluentChecksNeededForNoxRateCalc*** is true)

Set ***CurrentMhvComponentType*** to "CO2".
Set ***CurrentMhvDefaultParame3ter*** to "CO2X".
Set ***CurrentMhvParameter*** to "CO2C".
Set ***CurrentMhvParameterDescription*** to "CO2C".

Set ***CurrentMhvParameterStatus*** to true.

Else
Return result B.

"NOXC":

Set ***CurrentMhvComponentType*** to "NOX".
Set ***CurrentMhvDefaultParame3ter*** to "NOCX".
Set ***CurrentMhvParameter*** to "NOXC".
Set ***CurrentMhvParameterDescription*** to "NOXC".

Set ***CurrentMhvParameterStatus*** to true.

"O2C":

If (***O2DryChecksNeededForNoxrRateCalc*** is true) AND (***O2WetChecksNeededForNoxrRateCalc*** is NOT true)

Set ***CurrentMhvComponentType*** to "O2".
Set ***CurrentMhvDefaultParame3ter*** to "O2N".
Set ***CurrentMhvParameter*** to "O2D".
Set ***CurrentMhvParameterDescription*** to "O2 Dry".

Set ***CurrentMhvParameterStatus*** to true.

Else If (***O2DryChecksNeededForNoxrRateCalc*** is NOT true) AND (***O2WetChecksNeededForNoxrRateCalc*** is true)

Set ***CurrentMhvComponentType*** to "O2".
Set ***CurrentMhvDefaultParame3ter*** to "O2N".
Set ***CurrentMhvParameter*** to "O2W".
Set ***CurrentMhvParameterDescription*** to "O2 Wet".

Set ***CurrentMhvParameterStatus*** to true.

Else

Return result C.

Otherwise

Return result A.

Results:

Result	Response	Severity
A	MODC [modcCd] is invalid for MHV parameter [parameterCd].	Critical Error Level 1
B	You reported a [parameterCd] MHV record for MODC [modcCd], but [parameterCd] is not required for NOXR.	Critical Error Level 1
C	You reported a [parameterCd] MHV record for MODC [modcCd], but [parameterCd] is not required for NOXR.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ NOXR Unused P-PB Monitor Hourly Evaluation

Check Code: HOURMHV-30

Check Name: NOxR Primary/Primary Bypass: Component Check

Related Former Checks: HOURMHV-15

Applicability: CEM Check

Description: Validates the component for the MHV record.

Specifications:

Set ***MonitorHourlyComponentStatus*** to false.
Set ***MonitorHourlySystemStatus*** to false.

If (***CurrentMhvParameterStatus*** is true)

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.ComponentId is null)

Return result A.

Else If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.ComponentTypeCode does NOT equal ***CurrentMhvComponentType***)

Return result B.

Else

Set ***MonitorHourlyComponentStatus*** to true.

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.NotReportedNoxrSystemCount does NOT equal 1)

Return result C.

Else

Set ***MonitorHourlySystemStatus*** to true.

Results:

Result	Response	Severity
A	You did not reported a ComponentID in the MHV record for [parameterCd] and MODC [modcCd].	Critical Error Level 1
B	You reported [reportedComponentType] ComponentID [componentIdentifier] in the MHV record for [parameterCd] and MODC [modcCd], but a [expectedComponentType] component is expected.	Critical Error Level 1
C	You reported ComponentID [componentIdentifier] in the MHV record for [parameterCd] and MODC [modcCd]. This component was used to determine the reported value in the NOx emission rate DHV record and therefore cannot report MODC “47” or “48”.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ NOXR Unused P-PB Monitor Hourly Evaluation

Check Code: HOURMHV-31

Check Name: NOxR Primary/Primary Bypass: Determine Maximum Allowed Parameter Value

Related Former Checks: HOURMHV-18

Applicability: CEM Check

Description: Determines the maximum value allowed value for CO2C, NOXC or O2C for MODC 47.

Specifications:

Set ***CurrentMhvMaxMinValue*** to null.
Set ***CurrentNoxrPrimaryOrPrimaryBypassMhvMaxValueDescription*** to null

If (***CurrentMhvParameterStatus*** is true) AND (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.ModcCode equals "47")

When ***CurrentMhvParameter*** equals

"CO2C":

Set ***CurrentNoxrPrimaryOrPrimaryBypassMhvMaxValueDescription*** to "CO2 Span High Range".

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanCount equals 0)
Return result A.
Else If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanCount is greater than 1)
Return result B.
Else
If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanFullScaleRange is NOT null)
Set *MaxValue* to (2 *
CurrentNoxrPrimaryOrPrimaryBypassMhvRecord.HighSpanFullScaleRange).
Else
Set *MaxValue* to null.

If (*MaxValue* is null) OR (
CurrentNoxrPrimaryOrPrimaryBypassMhvRecord.HighSpanDefaultHighRange is not null and is greater than *MaxValue*)
Set *MaxValue* to
CurrentNoxrPrimaryOrPrimaryBypassMhvRecord.HighSpanDefaultHighRange.

If (*MaxValue* is NOT null AND is greater than 0)

Set ***CurrentMhvMaxMinValue*** to *MaxValue.*
Else
Return result C.

"NOXC":

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanCount equals 0) AND (
CurrentNoxrPrimaryOrPrimaryBypassMhvRecord.LowSpanCount equals 0)

Set ***CurrentNoxrPrimaryOrPrimaryBypassMhvMaxValueDescription*** to "NOX Span".
Return result D.

Else If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanCount is greater than 1) OR (
CurrentNoxrPrimaryOrPrimaryBypassMhvRecord.LowSpanCount is greater than 1)

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanCount is greater than 1) AND (
CurrentNoxrPrimaryOrPrimaryBypassMhvRecord.LowSpanCount equals 0)
Set ***CurrentNoxrPrimaryOrPrimaryBypassMhvMaxValueDescription*** to "NOX Span High Range".

Else If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanCount equals 0) AND (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.LowSpanCount is greater than 1)
Set ***CurrentNoxrPrimaryOrPrimaryBypassMhvMaxValueDescription*** to "NOX Span Low Range".
Else
Set ***CurrentNoxrPrimaryOrPrimaryBypassMhvMaxValueDescription*** to "NOX Span".

Return result E.

Else
If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanDefaultHighRange is not null)

Set ***CurrentNoxrPrimaryOrPrimaryBypassMhvMaxValueDescription*** to "NOX Span Low Range".

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.LowSpanFullScaleRange is NOT null)
Set *MaxValue* to (2 * ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.LowSpanFullScaleRange).
Else
Set *MaxValue* to null.

If (*MaxValue* is null) OR (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanDefaultHighRange is greater than *MaxValue*)

Set ***CurrentNoxrPrimaryOrPrimaryBypassMhvMaxValueDescription*** to "NOX Span High Range".
Set *MaxValue* to ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanDefaultHighRange.

Else
Set ***CurrentNoxrPrimaryOrPrimaryBypassMhvMaxValueDescription*** to "NOX Span High Range".

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanFullScaleRange is NOT null)
Set *MaxValue* to (2 * ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.HighSpanFullScaleRange).
Else
Set *MaxValue* to null.

If (*MaxValue* is NOT null AND is greater than 0)

Set ***CurrentMhvMaxMinValue*** to *MaxValue*.
Else
Return result F.

"O2D" or "O2W":

Set ***CurrentNoxrPrimaryOrPrimaryBypassMhvMaxValueDescription*** to "O2C NFS Missing Data Default".

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.MaxDefaultCount equals 0)
Return result G.
Else If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.MaxDefaultCount is greater than 1)
Return result H.
Else
Set ***CurrentMhvMaxMinValue*** to ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.MaxDefaultValue.

Results:

Result	Response	Severity
A	You did not report an applicable [description] record in your monitoring plan that is active for the hour.	Critical Error Level 1
B	You reported more than one applicable [description] record in your monitoring plan that is active for the hour.	Critical Error Level 1
C	The values reported in the applicable [description] record are invalid.	Critical Error Level 1
D	You did not report an applicable [description] record in your monitoring plan that is active for the hour.	Critical Error Level 1
E	You reported more than one applicable [description] record in your monitoring plan that is active for the hour.	Critical Error Level 1
F	The values reported in the applicable [description] record are invalid.	Critical Error Level 1
G	You did not report an applicable [description] record in your monitoring plan that is active for the hour.	Critical Error Level 1
H	You reported more than one applicable [description] record in your monitoring plan that is active for the hour.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOXR Unused P-PB Monitor Hourly Evaluation

Check Code: HOURMHV-32

Check Name: NOxR Primary/Primary Bypass:Extraneous Data Check

Related Former Checks:

Applicability: CEM Check

Description: Ensures that unexpected data is not reported.

Specifications:

Set ***CurrentMhvExtraneousFields*** to "".

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.AdjustedHourlyValue is NOT null)
Append "AdjustedHourlyValue" to ***CurrentMhvExtraneousFields***.

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.MositureBasis is NOT null)
Append "MositureBasis" to ***CurrentMhvExtraneousFields***.

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.MonitorSystemID is NOT null)
Append "MonitorSystemID" to ***CurrentMhvExtraneousFields***.

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.PercentAvailable is NOT null)
Append "PercentAvailable" to ***CurrentMhvExtraneousFields***.

If (***CurrentMhvExtraneousFields*** does NOT equal "")
Return result A.

Results:

Result	Response	Severity
A	You reported [extraneousFields] in the MHV record for [parameterCd] and MODC [modcCd]. This data should be blank.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOXR Unused P-PB Monitor Hourly Evaluation

Check Code: HOURMHV-33

Check Name: NOxR Primary/Primary Bypass:Unadjusted Value Check

Related Former Checks:

Applicability: CEM Check

Description: Checks the reported unadjusted hourly value.

Specifications:

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnadjustedHourlyValue is null)

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.ModcCode equals "47"

Return result A.

Otherwise

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.ModcCode equals "48")

Return result B.

Else If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnadjustedHourlyValue is less than 0.0)

Return result C.

Else If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnadjustedHourlyValue is NOT rounded to 1 decimal place)

Return result D.

Else If (***CurrentMhvParameter*** equals "CO2C") AND (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnadjustedHourlyValue is greater than 16.0)

Return result E.

Else If (***CurrentMhvParameter*** in set { "O2D", "O2W") AND (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnadjustedHourlyValue is greater than 22.0)

Return result F.

Else If (***CurrentMhvParameter*** equals "CO2C") AND (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnadjustedHourlyValue equals 0.0) AND (***CurrentHourlyOpRecord***.LoadRange is greater than 1)

Return result G.

Else If (***CurrentMhvMaxMinValue*** is NOT null) AND (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnadjustedHourlyValue is greater than ***CurrentMhvMaxMinValue***)

Return result H.

Results:

Result	Response	Severity
A	An UnadjustedHourlyValue is required for [parameterCd] with MODC [modcCd].	Critical Error Level 1
B	An UnadjustedHourlyValue is not appropriate for [parameterCd] with MODC [modcCd].	Critical Error Level 1
C	The reported UnadjustedHourlyValue of [unadjustedHourlyValue] in the MHV record with parameter [parameterCd] and MODC [modcCd] has a negative value.	Critical Error Level 1
D	The reported UnadjustedHourlyValue of [unadjustedHourlyValue] in the MHV record with parameter [parameterCd] and MODC [modcCd] is not rounded to one decimal place.	Critical Error Level 1
E	The reported concentration of [unadjustedHourlyValue] in the MHV record with parameter [parameterCd] and MODC [modcCd] cannot exceed 16%.	Critical Error Level 1
F	The reported concentration of [unadjustedHourlyValue] in the MHV record with parameter [parameterCd] and MODC [modcCd] cannot exceed 22%.	Critical Error Level 1
G	The UnadjustedHourlyValue of 0 was reported in the MHV record with parameter [parameterCd] and MODC [modcCd], but the LoadRange is greater than 1. Emissions for [parameterCd] should be greater than 0 when the unit (or stack) is operating at this load level.	Critical Error Level 1
H	Warning: The reported UnadjustedHourlyValue of [unadjustedHourlyValue] in the MHV record with parameter [parameterCd] and MODC [modcCd] is in excess of the maximum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update Span and/or Default values. You should investigate the cause of these exceedances and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOXR Unused P-PB Monitor Hourly Evaluation

Check Code: HOURMHV-34

Check Name: NOxR Primary/Primary Bypass:Primary Bypass Active Check

Related Former Checks:

Applicability: CEM Check

Description: Ensures that an active Primary Bypass NOx system exists for the current loaction and hour.

Specifications:

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.PrimaryBypassExistsIndicator does NOT equal 1)

Return result A.

Results:

Result	Response	Severity
A	An MHV record for [parameterCd] with MODC [modcCd] was reported, but reporting MODC [modcCd] is only appropriate when a primary bypass system exists at the unit.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ NOXR Unused P-PB Monitor Hourly Evaluation

Check Code: HOURMHV-35

Check Name: NOxR Primary/Primary Bypass:Missing Expected MHV Records Check

Related Former Checks:

Applicability: CEM Check

Description: Indicates whether expected used and unused NOXC and diluent MHV records are missing.

Specifications:

Set ***CurrentMhvMissing*** to "".

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.PrimaryBypassExistsIndicator equals 1)

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UsedNoxcCount equals 0)
Append "Used NOXC" to ***CurrentMhvMissing***.

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UsedDiluentCount equals 0)
Append "Used CO2C/O2C" to ***CurrentMhvMissing***.

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnusedNoxcCount equals 0)
Append "Unused NOXC" to ***CurrentMhvMissing***.

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnusedDiluentCount equals 0)
Append "Unused CO2C/O2C" to ***CurrentMhvMissing***.

If (***CurrentMhvMissing*** does NOT equal "")
Return result A.

Results:

Result	Response	Severity
A	A MHV record for NOXC or diluent (CO2C/O2C) included MODC "47" or "48", but [missingList] MHV was/were not reported. Reporting MODC "47" or "48" requires the reporting of both a NOXC and diluent MHV with the same "unused" MODC, either "47" or "48". It also requires the reporting of NOXC and diluent MHV that do not contain MODC "47" or "48".	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOXR Unused P-PB Monitor Hourly Evaluation

Check Code: HOURMHV-36

Check Name: NOxR Primary/Primary Bypass:Duplicate Expected MHV Records Check

Related Former Checks:

Applicability: CEM Check

Description: Indicates whether multiple expected used and unused NOXC and diluent MHV records were reported.

Specifications:

Set ***CurrentMhvDuplicate*** to "".

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.PrimaryBypassExistsIndicator equals 1)

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UsedNoxcCount is greater than 1)
Append "Used NOXC" to ***CurrentMhvDuplicate***.

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UsedDiluentCount is greater than 1)
Append "Used CO2C/O2C" to ***CurrentMhvDuplicate***.

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnusedNoxcCount is greater than 1)
Append "Unused NOXC" to ***CurrentMhvDuplicate***.

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnusedDiluentCount is greater than 1)
Append "Unused CO2C/O2C" to ***CurrentMhvDuplicate***.

If (***CurrentMhvDuplicate*** does NOT equal "")
Return result A.

Results:

Result	Response	Severity
A	A MHV record for NOXC or diluent (CO2C/O2C) included MODC "47" or "48", but duplicate MHV records were reported for [duplicateList]. Reporting MODC "47" or "48" requires the reporting of both a NOXC and diluent MHV with the same "unused" MODC, either "47" or "48". It also requires the reporting of NOXC and diluent MHV that do not contain MODC "47" or "48". However, only one record is required for each.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOXR Unused P-PB Monitor Hourly Evaluation

Check Code: HOURMHV-37

Check Name: NOxR Primary/Primary Bypass:Compare Unused MODC

Related Former Checks:

Applicability: CEM Check

Description: Ensures that the unused MODC for NOXC and CO2C/O2C MHV are the same.

Specifications:

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.PrimaryBypassExistsIndicator equals 1)

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnusedNoxcCount equals 1) AND (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnusedDiluentCount equals 1)

If (***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnusedNoxcModcCode does NOT equal ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.UnusedDiluentModcCode)

Return result A.

Results:

Result	Response	Severity
A	The MHV records for NOXC and a diluent (CO2C/O2C) reported MODC "47" or "48", but did not report the same MODC which is required.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOXR Unused P-PB Monitor Hourly Evaluation

Check Code: HOURMHV-38

Check Name: NOxR Primary/Primary Bypass:Update Supplemental Counts

Related Former Checks:

Applicability: CEM Check

Description: Performs System Op Supp Data (and QA Cert Event Supp Data) updates for the NOXR system that was not reported.

Performs Component Op Supp Data (and QA Cert Event Supp Data) updates for the component reported in the MODC 47 or 48 MHV record.

Performs Last QA Value Supp Data updates for the component reported in the MODC 47 record.

Specifications:

If ***DerivedHourlyChecksNeeded*** is true AND ***CurrentOperatingTime*** is greater than 0,

If (***MonitorHourlySystemStatus*** equals true)

Perform the updates in HOUROP-48 for ***SystemOperatingSuppDataDictionaryArray*** with:

1) *HourlyRecord*..MonitoringSystemID replaced by ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.NotReportedNoxMonitoringSystemID.
2) *HourlyRecord*..ModcCode replaced by ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.ModcCode.

If (***MonitorHourlySystemStatus*** equals true)

Perform the updates in HOUROP-49 for ***ComponentOperatingSuppDataDictionaryArray*** with:

1) *HourlyRecord*..ComponentID replaced by ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.ComponentID.
2) *HourlyRecord*..ModcCode replaced by ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.ModcCode.

Perform the updates in HOUROP-50 for ***LastQaValueSuppDataDictionaryArray*** with:

1) *HourlyRecord*..MonitoringSystemID replaced by null.
3) *HourlyRecord*..ModcCode replaced by ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.ModcCode.
4) *HourlyTypeCode* set to "MONITOR".
5) *MoistureBasis* set to ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.MoistureBasis.
6) *ComponentKey* set to ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.ComponentID.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ NOXR Unused P-PB Monitor Hourly Evaluation

Check Code: HOURMHV-39

Check Name: NOxR Primary/Primary Bypass:Set Not Reported NOXR QA Status Information

Related Former Checks:

Applicability: CEM Check

Description: Sets the check parameters needed to perform RATA status checks for the "not reported" NOX system for NOXR.

Specifications:

Set ***QaStatusComponentId*** to null.
Set ***QaStatusComponentIdentifier*** to null.
Set ***QaStatusComponentTypeCode*** to null.

Set ***QaStatusHourlyParameterCd*** to "NOXR".

Set ***QaStatusSystemDesignationCode*** to ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.NotReportedSystemDesignationCode.
Set ***QaStatusSystemId*** to ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.NotReportedSystemId.
Set ***QaStatusSystemIdentifier*** to ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.NotReportedSystemIdentifier.
Set ***QaStatusSystemTypeCode*** to ***CurrentNoxrPrimaryOrPrimaryBypassMhvRecord***.NotReportedSystemTypeCode.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB NOX RATA Status Initialization

Check Code: HOURMHV-40

Check Name: Initialize QA Status Evaluation for Flow Averaging Component

Related Former Checks:

Applicability: General Check

Description: Determines whether Daily Calibration, Daily Interference and Leak status checks need to run for components involved with Flow reported using Dual Flow (X-Pattern Flow) monitoring systems.

Specifications:

Set ***DailyCalStatusRequired*** to false
Set ***DailyIntStatusRequired*** to false
Set ***LeakStatusRequired*** to false

Set ***ApplicableComponentId*** to null
Set ***CurrentAnalyzerRangeUsed*** to null.
Set ***CurrentDailyCalStatus*** to null.
Set ***DualRangeStatus*** to false.
Set ***HighRangeComponentId*** to null.
Set ***LowRangeComponentId*** to null.

Set ***QaStatusComponentBeginDate*** to null.
Set ***QaStatusComponentBeginDatehour*** to null.
Set ***QaStatusComponentId*** to null
Set ***QaStatusComponentIdentifier*** to null
Set ***QaStatusComponentTypeCode*** to null
Set ***QaStatusPrimaryOrPrimaryBypassSystemId*** to null.

When:

1) ***CurrentMHVParameter*** equals "FLOW".
2) ***FlowAveragingComponentReecord*** is NOT null.
3) ***FlowAveragingComponentReecord***.ComponentId is NOT null.
4) ***MonitorHourlyComponentStatus*** is true.
5) ***CurrentMhvRecord***.ComponentId is null.
6) ***MonitorHourlyModcStatus*** is true.
7) And either:
   a) ***CurrentMhvRecord***.ModcCode in set {01, 02, 03, 53}
   OR
   a) ***CurrentMhvRecord***.ModcCode equals "20".
   b) ***CurrentMhvRecord***.UnadjustedHourlyValue is NOT null.
   c) ***CurrentMhvMaxMinValue*** is not null.

Then:

Set ***QaStatusComponentId*** to ***FlowAveragingComponentRecord***.ComponentId.
Set ***QaStatusComponentIdentifier*** to ***FlowAveragingComponentRecord***.ComponentIdentifier.
Set ***QaStatusComponentTypeCode*** to ***FlowAveragingComponentRecord***.ComponentTypeCode.

Set ***ApplicableComponentId*** to ***FlowAveragingComponentRecord***.ComponentId.

Set ***DailyCalStatusRequired*** to true.
Set ***DailyIntStatusRequired*** to true.
Set ***LeakStatusRequired*** to (***FlowAveragingComponentRecord***.SampleAcquistionMethodCd equals "DP").

Results:

Result	Response	Severity

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Flow Averaging Component Evaluation

Check Category:

Hourly Operating Data

Check Code: HOUROP-1

Check Name: Validate Single Operating Data record for hour

Related Former Checks:

Applicability: CEM Check

Description: This check will count the number of HourlyOperatingData records to ensure the existence of one unique record for the hour

Specifications:

Current Hourly Op Record = null
Unit Hourly Operational Status = false
Current Operating Time = null
Hourly Extraneous Fields = null
Count all HourlyOperatingData records with current MonitoringLocationID where
BeginHour = ***Current Hour*** AND
BeginDate = ***Current Date***

If count == 0
Derived Hourly Checks Needed = false
if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
If (***Current Entity Type*** == "Unit" OR ***LME HI Method*** is null)
If (***Reporting Period Operating*** == false AND ***Legacy Data Evaluation*** == true)
return result E
else
Locate ***Monitor Method*** records where the BeginDate/BeginHour is on or before the ***Current Date and Hour***, and the EndDate/EndHour is null or is on or after the ***Current Date and Hour***.

If found
return result A

else if count > 1

if (***Current Month*** is not April OR ***Annual Reporting Requirement*** == true)
Rpt Period CO2 Mass Reported Accumulator Array for the location = -1
Rpt Period CO2 Mass Calculated Accumulator Array for the location = -1
Rpt Period HI Reported Accumulator Array for the location = -1
Rpt Period HI Calculated Accumulator Array for the location = -1
Rpt Period NOx Rate Reported Accumulator Array for the location = -1
Rpt Period NOx Rate Calculated Accumulator Array for the location = -1
Rpt Period SO2 Mass Reported Accumulator Array for the location = -1
Rpt Period SO2 Mass Calculated Accumulator Array for the location = -1
Rpt Period NOx Mass Reported Accumulator Array for the location = -1
Rpt Period NOx Mass Calculated Accumulator Array for the location = -1
Rpt Period Op Time Accumulator Array for the location = -1
Rpt Period Op Hours Accumulator Array for the location = -1
Daily Op Time Accumulator Array for the location = -1

Derived Hourly Checks Needed = false
return result B

else if (***Current Entity Type*** <> "Unit" AND ***LME HI Method*** is not null)
Derived Hourly Checks Needed = false
return result D

else
Current Hourly Op Record = Unique HourlyOperatingData record
Current Operating Time = ***Current Hourly Op Record***.OperatingTime

```
If (First Day of Operation is null)
        First Day of Operation = Current Hourly Op Record.Date
        First Hour of Operation = Current Hourly Op Record.Hour

if Current Operating Time > 1.0 OR Current Operating Time < 0.0
        Derived Hourly Checks Needed = false

        if (Current Month is not April OR Annual Reporting Requirement == true)
                Rpt Period Op Time Accumulator Array for the location  = -1
                Rpt Period Op Hours Accumulator Array for the location  = -1
                Daily Op Time Accumulator Array for the location  = -1

        if (Current Entity Type = "Unit")
                Unit OpTime Accumulator = -1
        else
                Stack OpTime Accumulator = -1

        return result C
else
        Derived Hourly Checks Needed = true
        if Current Operating Time > 0.0
                Unit Hourly Operational Status = true

                if (Operating Date Array entry for this location does not contain Current Hourly Op Record.Date)
                        Add Current Hourly Op Record.Date to Operating Date Array entry for this location

                if (Current Month is not April OR Annual Reporting Requirement == true)
                        if (Rpt Period Op Hours Accumulator Array for this location is not null)
                                if (Rpt Period Op Hours Accumulator Array for this location >= 0)
                                        Rpt Period Op Hours Accumulator Array for this location = Rpt Period Op Hours
                                        Accumulator + 1
                        else
                                Rpt Period Op Hours Accumulator Array for this location = 1

                        if (Rpt Period Op Time Accumulator Array for this location is not null)
                                if (Rpt Period Op Time Accumulator Array for this location  >= 0)
                                        Rpt Period Op Time Accumulator Array for this location  = Rpt Period Op Time
                                        Accumulator + Current Hourly Op Record.OperatingTime
                        else
                                Rpt Period Op Time Accumulator Array for this location  = Current Hourly Op
                                Record.OperatingTime

                        If (Current Month is April)
                                if (April Op Hours Accumulator Array for this location is not null)
                                        April Op Hours Accumulator Array for this location = April Op Hours Accumulator +
                                        1
                                else
                                        April Op Hours Accumulator Array for this location = 1

                                if (April Op Time Accumulator Array for this location is not null)
                                        April Op Time Accumulator Array for this location  = April Op Time Accumulator +
                                        Current Hourly Op Record.OperatingTime
                                else
                                        April Op Time Accumulator Array for this location  = Current Hourly Op
                                        Record.OperatingTime
```

if (***Daily Op Time Accumulator Array*** for this location is not null)
    if (***Daily Op Time Accumulator Array*** for this location >= 0)
        Daily Op Time Accumulator Array for this location = ***Daily Op Time Accumulator*** + ***Current Hourly Op Record***.OperatingTime
else
    Daily Op Time Accumulator Array for this location = ***Current Hourly Op Record***.OperatingTime

if (***Last Day of Operation Array*** for the location is null OR is not equal to ***Current Date***)
    Last Day of Operation Array for the location = ***Current Date***
    if (***Rpt Period Op Days Accumulator Array*** for this location is not null)
        if (***Rpt Period Op Days Accumulator Array*** for this location >= 0)
            Rpt Period Op Days Accumulator Array for this location = ***Rpt Period Op Days Accumulator*** + 1
    else
        Rpt Period Op Days Accumulator Array for this location = 1

    If (***Current Month*** is April)
        if (***April Op Days Accumulator Array*** for this location is not null)
            April Op Days Accumulator Array for this location = ***April Op Days Accumulator*** + 1
        else
            April Op Days Accumulator Array for this location = 1

if (***Current Entity Type*** = "Unit")
    if (***Unit OpTime Accumulator*** >= 0)
        Unit OpTime Accumulator = ***Unit OpTime Accumulator*** + ***Current Hourly Op Record***.OperatingTime

    if ***Current Hourly Op Record***.OperatingTime > ***Max Unit OpTime***
        Max Unit OpTime = ***Current Hourly Op Record***.OperatingTime

else if (***Current Entity Type*** == "CS" OR ***Current Entity Type*** == "MS")
    if (***Stack OpTime Accumulator*** >= 0)
        Stack OpTime Accumulator = ***Stack OpTime Accumulator*** + ***Current Hourly Op Record***.OperatingTime

    if ***Current Hourly Op Record***.OperatingTime > ***Max Stack OpTime***
        Max Stack OpTime = ***Current Hourly Op Record***.OperatingTime

Results:

Result	Response	Severity
A	You did not report an Hourly Operating record for the hour.	Critical Error Level 1
B	You reported more than one Hourly Operating records for the hour. There will be no further evaluation of the reported emissions data for this hour.	Critical Error Level 1
C	The OperatingTime reported in the Hourly Operating record is invalid. This value must be between 0 and 1. There will be no further evaluation of the reported emissions data for this hour.	Critical Error Level 1
D	You reported an invalid Hourly Operating record. Only the units in an LME monitoring configuration should report this record. There will be no further evaluation of the reported emissions data for this hour.	Critical Error Level 1
E	You did not report an Hourly Operating record for the hour. Although this was acceptable for legacy data during a non-operating quarter, it is not allowed in ECMPS.	Fatal

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-2

Check Name: Count Flow, O2, and Heat Input records

Related Former Checks:

Applicability: CEM Check

Description: Determine the number of Stack Flow Monitor Hourly, H2O Monitor Hourly, H2O Derived Hourly, Heat Input Derived Hourly, and O2 Wet and Dry Hourly Records for the current location and current hour

Specifications:

Flow Monitor Hourly Count = count of MonitorHourlyValueData records with parameter FLOW where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

O2 Wet Monitor Hourly Count = count of MonitorHourlyValueData records with ParameterCode = "O2C" AND MoistureBasis = "W" where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

O2 Dry Monitor Hourly Count = count of MonitorHourlyValueData records with ParameterCode = "O2C" AND MoistureBasis = "D" where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

O2 Null Monitor Hourly Count = count of MonitorHourlyValueData records with ParameterCode = "O2C" AND MoistureBasis is NULL where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour
if ***O2 Null Monitor Hourly Count*** == 1
Current O2 Null Monitor Hourly Record = the single matching record

Heat Input Derived Hourly Count = count of DerivedHourlyValueData records with ParameterCode equal to "HI" where
Current Date = DerivedHourlyValueData.Date and
Current Hour = DerivedHourlyValueData.Hour

// ***O2 Needed To Support Heat Input*** = false

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-3

Check Name: Initialize Location Variables for the Hour

Related Former Checks:

Applicability: General Check

Description: Looks up information about the current unit - initially whether or not it is a Peaking Unit

Specifications:

Current Unit Is Peaking = false
Current Unit Is ARP = false
Special Fuel Burned = false
FC Factor Needed = false
FD Factor Needed = false
FW Factor Needed = false
Moisture Needed = false
H2O Missing Data Approach = null
Flow Monitor Hourly Checks Needed = false
Flow Needed For Part 75 = false
Flow MHV Optionally Allowed = false
Current MHV Parameter = null
Current DHV Parameter = null
Current DHV Record Valid = false
Current DHV Record = null
Current DHV Method = null
Current DHV System Type = null
Current DHV HBHA Value = null
CO2 Conc Checks Needed for CO2 Mass Calc = false
CO2 Conc Checks Needed for Heat Input = false
CO2 Diluent Checks Needed for NOx Rate Calc = false
O2 Dry Checks Needed for Heat Input = false
O2 Dry Checks Needed for NOx Rate Calc = false
O2 Wet Checks Needed for Heat Input = false
O2 Wet Checks Needed for NOx Rate Calc = false
CO2 Diluent Needed for MATS = false
CO2 Diluent Needed for MATS Calculation = false
O2 Dry Needed for MATS = false
O2 Dry Needed for MATS Calculation = false
O2 Wet Needed for MATS = false
O2 Wet Needed for MATS Calculation = false
Linearity Status Required = false
AppendixEStatusRequired = false
RATAStatusRequired = false
Current RATA Status = null
CurrentHourlyRecordforRATAStatus = null
RATAStatusBAF = null
Daily Cal Status Required = false
CO2 RATA Required = false
HI Measure Code = null.
NOXR Measure Code = null.
F2L Status Required = false
Daily Int Status Required = false
Leak Status Required = false

CO2C MHV MODC = null
H2O DHV MODC = null
H2O MHV MODC = null
O2 Dry MODC = null

O2 Wet MODC = null

SO2 HPFF Exists = false
CO2 HPFF Exists = false
HI HPFF Exists = false

If the StackPipeID of the monitoring location begins with "CS",
    set ***Current Entity Type*** = "CS"

else if the StackPipeID of the monitoring location begins with "CP",
    set ***Current Entity Type*** = "CP"

else if the StackPipeID of the monitoring location begins with "MS",
    set ***Current Entity Type*** = "MS"

else if the StackPipeID of the monitoring location begins with "MP",
    set ***Current Entity Type*** = "MP"

else
    set ***Current Entity Type*** = "Unit"

if ***Current Entity Type*** = "Unit"

    Mon Qual Record Count = Find Monitor Qualification Records by Hour where
        MonitoringLocationId = ***Current Location***
        Mon Qual Record.QualTypecode == "PK" OR *Mon Qual Record*.QualTypecode == "SK"

    if *Mon Qual Record Count > 0*
        Current Unit Is Peaking = true

else if ***Current Entity Type*** = "CP"
    find all entries in UnitStackConfiguration table where
        UnitStackConfiguration.StackPipeId = the StackPipeId for this pipe
    for each matching record
        set *Assoc Unit* = UnitStackConfiguration.UnitId
        Mon Qual Record Count = Find MonitorQualification Records by Hour where
            MonitoringLocationId = *Assoc Unit*
            Mon Qual Record.QualTypecode == "PK" OR *Mon Qual Record*.QualTypecode == "SK"
        if *Mon Qual Record Count* is not null
            Current Unit Is Peaking = true

Set ***CurrentUnitIsArp*** to false.
Set ***So2cIsOnlyForMats*** to false.

Locate records in ***LocationProgramRecordsByHourLocation*** where:

1) EmissionRecordingDate is NOT null and is on or prior to ***CurrentOperatingDate***,
2) Or, EmissionRecordingDate is null, UnitMonitorCertBeginDate is NOT null, and UnitMonitorCertBeginDate is on or prior to ***CurrentOperatingDate***.

If the located ***LocationProgramRecordsByHourLocation*** include a record where ProgramCode is equal to "ARP" AND Class is equal to "P1" or "P2",
    Set ***CurrentUnitIsArp*** to true.

If ***CurrentUnitIsArp*** is equal to false

If the located ***LocationProgramRecordsByHourLocation*** include a record where ProgramCode is equal to "MATS" and Class is equal to "A", AND does NOT include a record where ProgramCode is NOT equal to "MATS" AND is in ***ProgramRequiresSo2SystemCertificationList*** and Class is equal to "A",
Set ***So2cIsOnlyForMats*** to true.

Set ***EarliestLocationReportDate*** = ***CurrentMonitorPlanLocationRecord***.EarliestReportDate

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-4

Check Name: Verify SO2 Monitor Method Active During Current Hour

Related Former Checks:

Applicability: General Check

Description: This check tests to see if the "SO2" Monitoring method is defined at this location. If so, this method is checked to ensure that the current hour being evaluated is within the window defined by the start and end times for the SO2 method.

Specifications:

If ***(Derived Hourly Checks Needed)***
SO2 Monitor Method Record = null
SO2 CEM Method Active For Hour = false
SO2 App D Method Active For Hour = false
SO2 F23 Method Active For Hour = false
SO2 Method Code = null
SO2 Fuel Specific Missing Data = false
SO2 Bypass Code = null

SO2 Method Count = Active records in MonitoringMethodData for the location where
ParameterCode = "SO2" or "SO2M"

if (***SO2 Method Count*** > 1)
return result A
else if ***SO2 Method Count*** == 1

SO2 Monitor Method Record = the single matched record
SO2 Method Code = ***SO2 Monitor Method Record.***MethodCode

if (***LME HI Method*** is not null AND ***SO2 Method Code*** <> "LME")
return result B
else
if (***SO2 Monitor Method Record.***SubDataCode begins with "FSP75")
SO2 Fuel Specific Missing Data = true
SO2 Bypass Code = ***SO2 Monitor Method Record.***BypassApproachCode

Expected Summary Value SO2 Array for this location = true

if (***SO2 Monitor Method Record.***MethodCode == "CEM")
SO2 CEM Method Active For Hour = true
else if ***SO2 Monitor Method Record.***MethodCode == "F23")
SO2 F23 Method Active For Hour = true
else if (***SO2 Monitor Method Record.***MethodCode == "AD")
SO2 App D Method Active For Hour = true

Results:

Result	Response	Severity
A	You have reported more than one active SO2 Method record in your monitoring plan for this hour.	Critical Error Level 1
B	You reported an invalid [param] method for a location that is part of a configuration of LME units.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-5

Check Name: Determine H2O Method

Related Former Checks:

Applicability:

Description: The H2O Monitor Method must be known prior to category-level checks for H2O Derived and H2O Monitor

Specifications:

if ***Derived Hourly Checks Needed***
    H2O Method Code = null
    H2O Default Value = null
    H2O Default Max Value = null
    H2O Default Min Value = null
    Current Hourly H2O Table Reference = null
    H2O Fuel Specific Missing Data = false

    H2O Reported Value = null
    H2O Method Count = Active records in MonitoringMethodData for the location where
        ParameterCode = "H2O"

    H2O Derived Hourly Count = count of DerivedHourlyValueData where ParameterCode = "H2O" for current hour
    H2O Monitor Hourly Count = count of MonitorHourlyValueData where ParameterCode = "H2O" for current hour
    if (***H2O Method Count*** > 2
        return result A
    else if (***H2O Method Count*** == 2)
        If (***H2O Derived Hourly Count*** + ***H2O Monitor Hourly Count*** > 0)
            If (***H2O Derived Hourly Count*** == 1 AND ***H2O Monitor Hourly Count*** == 0 AND the MethodCode in one of the matched records is equal to "MWD")
                Current Hourly H2O Table Reference = DerivedHourlyValueData where ParameterCode = "H2O"
                H2O Reported Value = ***Current Hourly H2O Table Reference***.AdjustedHourlyValue
                H2O Method Code = "MWD"

            else if ***(H2O Derived Hourly Count*** == 0 AND ***H2O Monitor Hourly Count*** == 1 AND the MethodCode in one (but not both) of the matched records is equal to "MTB" or "MMS")
                Current Hourly H2O Table Reference = MonitorHourlyValueData where ParameterCode = "H2O"
                H2O Reported Value = ***Current Hourly H2O Table Reference***.UnadjustedHourlyValue
                if the MethodCode in the matched record == "MMS"
                    H2O Method Code = "MMS"
                else
                    H2O Method Code = "MTB"

            else
                return result A
    else if ***H2O Method Count*** == 1

        H2O Monitor Method Record = the single matched record
        H2O Method Code = ***H2O Monitor Method Record***.MethodCode
        if (***H2O Monitor Method Record***.SubDataCode begins with "FSP75")
            H2O Fuel Specific Missing Data = true

        if (***H2O Method Code*** == 'MDF')
            H2O Default Record Count = count active MonitoringDefaultData Records for the location where ParameterCd = 'H2O'
            if (***H2O Default Record Count*** == 0)
                return result B
            else if (***H2O Default Record Count*** >1)
                if (***H2O Derived Hourly Count*** == 1)

Current Hourly H2O Table Reference = DerivedHourlyValueData where ParameterCode = "H2O"

H2O Default Max Value = Highest DefaultValue field from active MonitoringDefaultData record for location where ParameterCd = 'H2O'

H2O Default Min Value = Lowest DefaultValue field from active MonitoringDefaultData record for location where ParameterCd = 'H2O'

If (***H2O Default Max Value*** <= 0 OR ***H2O Default Min Value*** <= 0 OR ***H2O Default Max Value*** >= 100 OR ***H2O Default Min Value*** >= 100)
return result C

else

H2O Default Value = DefaultValue field from active MonitoringDefaultData record for location where ParameterCd = 'H2O'

If (***H2O Default Value*** <= 0 OR ***H2O Default Value*** >= 100)
return result C

else if (***H2O Method Code*** == "MWD")
if (***H2O Derived Hourly Count*** == 1)
Current Hourly H2O Table Reference = DerivedHourlyValueData where ParameterCode = "H2O"
H2O Reported Value = ***Current Hourly H2O Table Reference***.AdjustedHourlyValue

else if (***H2O Method Code*** == "MMS" OR ***H2O Method Code*** == "MTB")
if (***H2O Monitor Hourly Count*** == 1)
Current Hourly H2O Table Reference = MonitorHourlyValueData where ParameterCode = "H2O"
H2O Reported Value = ***Current Hourly H2O Table Reference***.UnadjustedHourlyValue

Results:

Result	Response	Severity
A	You have reported more than one active H2O Method record in your monitoring plan for this hour.	Critical Error Level 1
B	You reported an H2O MethodCode of MDF, but you did not report an active H2O default record in your monitoring plan for the hour.	Critical Error Level 1
C	The DefaultValue reported in the active H2O default record in your monitoring plan is invalid.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-6

Check Name: Verify NOx Rate Monitor Method

Related Former Checks:

Applicability: General Check

Description: This check tests to see if the "NOx Rate" Monitoring method is defined at this location. If so, this method is checked to ensure that the current hour being evaluated is within the window defined by the start and end times for the NOx method.

Specifications:

If ***(Derived Hourly Checks Needed)***
NOx Rate Bypass Code = null
NOx Rate Fuel Specific Missing Data = false
Current NOx Rate Monitor Method Record = null
Current NOx Rate Method Code = null
NOx Rate Method Count = Active records in MonitoringMethodData for the location where
ParameterCode = "NOXR"

if (***NOx Rate Method Count*** > 1)
return result A
else if ***NOx Rate Method Count*** == 1

if (***LME HI Method*** is not null)
return result B
else
Current NOx Rate Monitor Method Record = the single matched record
Current NOx Rate Method Code = ***Current NOx Rate Monitor Method Record.***MethodCode
NOx Rate Bypass Code = ***Current NOx Rate Monitor Method Record***.BypassApproachCode
if (***Current NOx Rate Monitor Method Record***.SubDataCode begins with "FSP75")
NOx Rate Fuel Specific Missing Data = true

If (***Current Unit is ARP*** == true)
Expected Summary Value NOx Rate Array for this location = true

Results:

Result	Response	Severity
A	You reported more than one active NOXR Method record in your monitoring plan for this hour.	Critical Error Level 1
B	You reported a [param] method, which is not valid for a location that is part of a configuration of LME units.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-7

Check Name: Verify NOx Mass Monitor Method Record

Related Former Checks:

Applicability: CEM Check

Description: Finds the Monitor Method record for NOx Mass and stores it for later reference

Specifications:

If ***(Derived Hourly Checks Needed)***
- ***Current NOx Mass Monitor Method Record*** = null
- ***NOx Mass Method Active For Hour*** = false
- ***NOx Mass Monitor Method Code*** = null
- ***NOx Mass Bypass Code*** = null
- ***NOx Mass Fuel Specific Missing Data*** = false

NOx Mass Method Count = Active records in MonitoringMethodData for the location where
ParameterCode = "NOX" or "NOXM"

if (***NOx Mass Method Count*** > 1)
return result A
else if ***NOx Mass Method Count*** == 1
Current NOx Mass Monitor Method Record = the single matched record
NOx Mass Monitor Method Code = ***NOx Mass Monitor Method Record.***MethodCode

if (***LME HI Method*** is not null AND ***NOx Mass Monitor Method Code*** <> "LME")
return result B
else

Expected Summary Value NOx Mass Array for this location = true
NOx Mass Bypass Code = ***Current NOx Mass Monitor Method Record***.BypassApproachCode
if (***Current NOx Mass Monitor Method Record***.SubDataCode begins with "FSP75")
NOx Mass Fuel Specific Missing Data = true

if (***NOx Mass Monitor Method Record.***MethodCode == "CEM" OR
NOx Mass Monitor Method Record.MethodCode == "NOXR" OR
NOx Mass Monitor Method Record.MethodCode == "CEMNOXR" OR
NOx Mass Monitor Method Record.MethodCode == "AMS")
NOx Mass Method Active For Hour = true

If (***NOx Mass Monitor Method Code*** == "LME" AND ***Current Unit is ARP*** == true)
Expected Summary Value NOx Rate Array for this location = true

Results:

Result	Response	Severity
A	You reported more than one active NOX Method record in your monitoring plan for this hour.	Critical Error Level 1
B	You reported an invalid [param] method for a location that is part of a configuration of LME units.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-8

Check Name: Verify CO2 Method Active During Current Hour

Related Former Checks:

Applicability: CEM Check

Description: Looks in the MonitoringMethod table to verify that a CO2 Mass Monitoring Method is active for the current hour at the current location

Specifications:

If ***(Derived Hourly Checks Needed)***
CO2 Monitor Method Record = null
CO2 CEM Method Active For Hour = false
CO2 App D Method Active For Hour = false
CO2 Fuel Specific Missing Data = false
CO2 Method Code = null

// AD and CEMs are possible method codes
CO2 Method Count = Active records in MonitoringMethodData for the location where
ParameterCode = "CO2" or "CO2M"

if (***CO2 Method Count*** > 1)
return result A
else if ***CO2 Method Count*** == 1
CO2 Monitor Method Record = the single matched record
CO2 Method Code = ***CO2 Monitor Method Record***.MethodCode

if (***LME HI Method*** is not null and ***CO2 Method Code*** is not equal to "LME")
return result B
else
if (***Current CO2 Monitor Method Record***.SubDataCode begins with "FSP75")
CO2 Fuel Specific Missing Data = true

if (***CO2 Monitor Method Record.***MethodCode == "CEM")
CO2 CEM Method Active For Hour = true
else if (***CO2 Monitor Method Record.***MethodCode == "AD")
CO2 App D Method Active For Hour = true

Expected Summary Value CO2 Array for this location = true

Results:

Result	Response	Severity
A	You have reported more than one active CO2 Method record in your monitoring plan for this hour.	Critical Error Level 1
B	You reported an invalid [param] method for a location that is part of a configuration of LME units.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-9

Check Name: Verify Heat Input Method Active During Current hour

Related Former Checks:

Applicability: General Check

Description: Verifies that a single method is defined for Heat Input during the Current Hour

Specifications:

If ***(Derived Hourly Checks Needed)***
Heat Input Monitor Method Record = null
Heat Input Fuel Specific Missing Data = false
Heat Input Method Code = null
Heat Input CEM Method Active For Hour = false
Heat Input App D Method Active For Hour = false

Heat Input Method Count = Active records in MonitoringMethodData for the location where
ParameterCode begins with "HI"

if (***Heat Input Method Count*** > 1)
return result A

else if (***LME HI Method*** is not null AND (***Heat Input Method Count*** == 0 OR ParameterCode in the matched record is not equal to "HIT"))
return result B

else if (***Heat Input Method Count*** == 1)
Heat Input Monitor Method Record = the single matched record
Heat Input Method Code = ***Heat Input Monitor Method Record***.MethodCode
LME HI Substitute Data Code = ***Heat Input Monitor Method Record***.SubstituteDataCode

if (***Heat Input Monitor Method Record***.SubDataCode begins with "FSP75")
Heat Input Fuel Specific Missing Data = true

if (***Heat Input Monitor Method Record.***MethodCode == "CEM")
Heat Input CEM Method Active For Hour = true
else if (***Heat Input Monitor Method Record.***MethodCode == "AD" OR ***Heat Input Monitor Method Record.***MethodCode == "ADCALC")
Heat Input App D Method Active For Hour = true

If (***Heat Input Monitor Method Record.***MethodCode <> "EXP")
Expected Summary Value HI Array for this location = true

if (***Heat Input Monitor Method Record***.ParameterCode == "HI")
Apportionment HI Method Array for this location = ***Heat Input Method Code***

Results:

Result	Response	Severity
A	You have reported more than one active HI Method record in your monitoring plan for this hour.	Critical Error Level 1
B	You did not report an HIT Method record for this location in your monitoring plan, which is required for all locations that are part of a configuration of LME units.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-17

Check Name: Verify Single SO2 Derived Hourly Data Record

Related Former Checks:

Applicability: CEM Check

Description: This check scans the DerivedHourlyValueData records to ensure that a single record containing SO2 derived values is reported for the current hour

Specifications:

If (***Derived Hourly Checks Needed*** == true)
SO2 Derived Checks Needed = false
SO2M Derived Checks Needed = false
Current SO2 Derived Hourly Record = null
F23 Default Max Value = null
F23 Default Min Value = null
F23 Default Value = null

SO2 Derived Hourly Count = count of DerivedHourlyValueData records with ParameterCode = "SO2" or "SO2M" where
Current Date = DerivedHourlyValueData.Date and
Current Hour = DerivedHourlyValueData.Hour

If ***Current Hourly Op Record***.OperatingTime > 0
If (***SO2 Derived Hourly Count*** == 0 AND ***SO2 Method Code*** is not null)
If (***SO2 Method Code*** == "AD")
If (***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil*** > 0)
return result A
else
return result A
Else if (***SO2 Derived Hourly Count*** > 0 AND ***SO2 Method Code*** is null)
Rpt Period SO2 Mass Reported Accumulator Array for the location = -1
Rpt Period SO2 Mass Calculated Accumulator Array for the location = -1
return result B
Else if (***SO2 Derived Hourly Count*** > 1)
Rpt Period SO2 Mass Reported Accumulator Array for the location = -1
Rpt Period SO2 Mass Calculated Accumulator Array for the location = -1
return result C
Else if (***SO2 Derived Hourly Count*** > 0 AND ***SO2 Method Code*** == "AD" AND ***Hourly Fuel Flow Count for Gas*** + ***Hourly Fuel Flow Count for Oil*** == 0)
Rpt Period SO2 Mass Reported Accumulator Array for the location = -1
Rpt Period SO2 Mass Calculated Accumulator Array for the location = -1
return result G
Else if (***SO2 Derived Hourly Count*** == 1)

Current SO2 Derived Hourly Record = DerivedHourlyValueData rec matching with param SO2 or SO2M where ***Current Date*** = DerivedHourlyValueData.Date and ***Current Hour*** = DerivedHourlyValueData.Hour

If (***LME HI Method*** is not null)
If (***SO2 Method Code*** == "LME")

If (***Current SO2 Derived Hourly Record***.ParameterCode == "SO2M")
SO2M Derived Checks Needed = true
else
Rpt Period SO2 Mass Reported Accumulator Array for the location = -1
Rpt Period SO2 Mass Calculated Accumulator Array for the location = -1
return result H

else

Rpt Period SO2 Mass Reported Accumulator Array for the location = -1
Rpt Period SO2 Mass Calculated Accumulator Array for the location = -1

else

If (***Current SO2 Derived Hourly Record***.ParameterCode == "SO2M")
Rpt Period SO2 Mass Reported Accumulator Array for the location = -1
Rpt Period SO2 Mass Calculated Accumulator Array for the location = -1
return result H

else
SO2 Derived Checks Needed = true

If (***SO2 Method Code*** in set {CEMF23,AMS})

if (***SO2 Method Code*** == "CEMF23")
SO2 CEM Method Active For Hour = true

if (***Current SO2 Derived Hourly Record***.FormulaIdentifier is not null)

SO2 Formula Record = MonitorFormulaData record where
MonitorFormulaData.FormulaID = ***Current SO2 Derived Hourly Record.***FormulaIdentifier

If (SO2 Formula Record is not null)

If (SO2 Formula Record.ParameterCode == "SO2")

if (***SO2 Method Code*** == "CEMF23")
If (SO2 Formula Record.EquationCode == "F-23")
SO2 F23 Method Active For Hour = true
SO2 CEM Method Active For Hour = false

if (***SO2 Method Code*** == "AMS")
If (SO2 Formula Record.EquationCode in set
{F-1,F-2})
SO2 Method Code == "CEM"
SO2 CEM Method Active For Hour = true

If (***SO2 F23 Method Active For Hour*** == true)

F23 Default Record Count = count active MonitoringDefaultData Records for the location where ParameterCd = 'SO2R' and DefaultPurposeCd = 'F23'

if (F23 Default Record Count == 0)
return result D
else if (F23 Default Record Count > 1)
F23 Default Max Value = Highest DefaultValue field from active MonitoringDefaultData record for location where ParameterCd = 'SO2R' and DefaultPurposeCd = 'F23'

F23 Default Min Value = Lowest DefaultValue field from active MonitoringDefaultData record for location where ParameterCd = 'SO2R' and DefaultPurposeCd = 'F23'

If (***F23 Default Max Value*** <= 0 OR ***F23 Default Min Value*** <= 0)
return result E

else

F23 Default Value = DefaultValue field from active MonitoringDefaultData record for location where ParameterCd = 'SO2R' and DefaultPurposeCd = 'F23'

If (***F23 Default Value*** <= 0)
return result E

else

If ***SO2 Derived Hourly Count*** > 0
Return result F

Results:

Result	Response	Severity
A	You did not report a DHV record for SO2 (or SO2M) for the hour.	Critical Error Level 1
B	You reported a DHV record for SO2 (or SO2M), but you did not report an active SO2 method record in your monitoring plan for the hour.	Critical Error Level 1
C	You reported more than one DHV records for SO2 (or SO2M) for the hour.	Critical Error Level 1
D	You did not report an active SO2R default record in your monitoring plan for use in F23 calculation for the hour.	Critical Error Level 1
E	The DefaultValue reported in the active SO2R F23 default record in your monitoring plan is invalid.	Critical Error Level 1
F	You reported a DHV record for SO2 (or SO2M), but this is not appropriate for a non-operating hour.	Critical Error Level 1
G	You reported a DHV record for [param], but you did not report any Hourly Fuel Flow records at the location.	Critical Error Level 1
H	The ParameterCode reported in the DHV record does not match the ParameterCode in the Method record in your monitoring plan used to determine [eparam].	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-18

Check Name: Verify Single SO2 Concentration record

Related Former Checks:

Applicability: CEM Check

Description: Counts all SO2 Concentration records (MonitorHourlyValueData records with "SO2C" ParameterCode) for the current hour and outputs appropriate responses if count does not match expectations

Specifications:

Current SO2 Monitor Hourly Record = null
SO2 Monitor Hourly Count = count of MonitorHourlyValueDate records with param "SO2C" where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour
If ***Unit Hourly Operational Status*** = true
If (***SO2 Monitor Hourly Count*** >0 AND ***SO2 CEM Method Active For Hour*** == false AND ***MATS SO2C Needed*** == false)

Return result A
Else if (***SO2 Monitor Hourly Count*** >1)
Return result B
Else if (***SO2 Monitor Hourly Count*** ==1)
Current SO2 Monitor Hourly Record = MonitorHourlyValueData rec with param SO2C where
CurrentDate = MonitorHourlyValueData.Date and
CurrentHour = MonitorHourlyValueData.Hour
Else
if (***SO2 Monitor Hourly Count*** > 0)
return result C

Results:

Result	Response	Severity
A	You reported an MHV record for SO2C, but you did not report an active SO2 method record in your monitoring plan for the hour.	Critical Error Level 1
B	You reported more than one MHV record for SO2C for the hour.	Critical Error Level 1
C	You reported an MHV record for SO2C, but this is not appropriate for a non-operating hour.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-19

Check Name: Verify Single NOx Concentration Record

Related Former Checks:

Applicability: CEM Check

Description: Counts all NOx Rate Concentration records (MonitorHourlyValueData records with "NOXR" ParameterCode) for the current hour and outputs appropriate responses if count does not match expectations

Specifications:

Current NOx Conc Monitor Hourly Record = null

NOx Conc Monitor Hourly Count = count of MonitorHourlyValueDate records with param "NOXC" where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

if ***Unit Hourly Operational Status*** = true
if (***NOx Conc Monitor Hourly Count*** >1)
Return result A
Else if (***NOx Conc Monitor Hourly Count*** ==1)
If (***NOx Mass Monitor Method Code*** in {CEM, CEMNOXR, AMS} OR ***Current NOx Rate Method Code*** in {CEM,AMS})
Current NOx Conc Monitor Hourly Record = MonitorHourlyValueData rec with param "NOXC" where
CurrentDate = MonitorHourlyValueData.Date and
CurrentHour = MonitorHourlyValueData.Hour
else
return result B

Else
if (***NOx Conc Monitor Hourly Count*** > 0)
return result C

Results:

Result	Response	Severity
A	You reported more than one MHV record for NOXC for the hour.	Critical Error Level 1
B	You reported an MHV record for NOXC, but you did not report an appropriate NOXR or NOX method record in your monitoring plan for the hour.	Critical Error Level 1
C	You reported an MHV record for NOXC, but this is not appropriate for a non-operating hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-20

Check Name: Verify Single NOx Rate Derived Hourly Record

Related Former Checks:

Applicability: CEM Check

Description: This check scans the DerivedHourlyValueData records to ensure that a single record containing SO2 derived values is reported for the current hour

Specifications:

Current NOXR Derived Hourly Record = null
NOXR Derived Hourly Checks Needed = null
NOXR Derived Hourly Count = null
NOXR Has Measured DHV MODC = null

If (***Derived Hourly Checks Needed*** == true)
NOx Rate Derived Checks Needed = false
NOx Rate Derived Hourly Count = count of DerivedHourlyValueData records with ParameterCode = "NOXR" where
Current Date = DerivedHourlyValueData.Date and
Current Hour = DerivedHourlyValueData.Hour

If ***Current Hourly Op Record***.OperatingTime > 0
If (***NOx Rate Derived Hourly Count*** == 0 AND ***Current NOx Rate Method Code*** is not null)
Return result A
Else if (***NOx Rate Derived Hourly Count*** > 0 AND ***Current NOx Rate Method Code*** is NULL)
Rpt Period NOx Rate Reported Accumulator Array for the location = -1
Rpt Period NOx Rate Calculated Accumulator Array for the location = -1
Return result B
Else if (***NOx Rate Derived Hourly Count*** > 1)
Rpt Period NOx Rate Reported Accumulator Array for the location = -1
Rpt Period NOx Rate Calculated Accumulator Array for the location = -1
Return result C
Else if (***NOx Rate Derived Hourly Count*** == 1)

Current NOx Rate Derived Hourly Record = DerivedHourlyValueData rec matching param NOXR
NOx Rate Derived Checks Needed = true
Apportionment NOXR Method Array at this location = ***Current NOx Rate Method Code***
NOXR Has Measured DHV MODC = (***Current NOx Rate Derived Hourly Record***.MODCCode in set { 01, 02, 03, 04, 05, 14, 21, 22, 53, 54 })

If (***Current NOx Rate Method Code*** == "AMS")

if (***Current NOx Rate Derived Hourly Record***.FormulaIdentifier is null)
if ***Current NOx Rate Derived Hourly Record***.MODCCode is not null)
Current NOx Rate Method Code = "CEM"

else
NOXR Formula Record = MonitorFormulaData record where
MonitorFormulaData.FormulaID = ***Current NOx Rate Derived Hourly Record.***FormulaIdentifier

If (NOXR Formula Record is not null)

If (NOXR Formula Record.ParameterCode == "NOXR" AND NOXR Formula Record.EquationCode in set {F-5,F-6,19-1,19-2,19-3,19-3D,19-4,19-5,19-5D,19-6,19-7,19-8,19-9})
Current NOx Rate Method Code = "CEM"

else

If ***NOx Rate Derived Hourly Count*** > 0

Return result D

Results:

Result	Response	Severity
A	You did not report a DHV record for NOXR for the hour.	Critical Error Level 1
B	You reported a DHV record for NOXR, but you did not report an active NOXR method record in your monitoring plan for the hour.	Critical Error Level 1
C	You reported more than one DHV record for NOXR for the hour.	Critical Error Level 1
D	You reported a DHV record for NOXR, but this is not appropriate for a non-operating hour.	Critical Error Level 1
E	This check result is obsolete.	No Errors

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-21

Check Name: Verify Single NOx Mass Derived Hourly Record

Related Former Checks:

Applicability: CEM Check

Description: Counts number of NOx Mass DerivedHourlyValue records active during the current hour and compares this count with the Monitor Method records indicating the need for this data

Specifications:

If (***Derived Hourly Checks Needed*** == true)
NOx Mass Derived Checks Needed = false
NOXM Derived Checks Needed = false
Current NOx Mass Derived Hourly Record = null

NOx Mass Derived Hourly Count = count of DerivedHourlyValueData records with ParameterCode = "NOX" or "NOXM" where
Current Date = DerivedHourlyValueData.Date and
Current Hour = DerivedHourlyValueData.Hour

If ***Current Hourly Op Record***.OperatingTime > 0
If (***NOx Mass Derived Hourly Count*** == 0 AND (***NOx Mass Method Active For Hour*** == true OR ***NOx Mass Monitor Method Code*** == "LME"))
Return result A

Else if (***NOx Mass Derived Hourly Count*** > 0 AND ***NOx Mass Method Active For Hour*** == false AND ***NOx Mass Monitor Method Code*** <> "LME")

Rpt Period NOx Mass Reported Accumulator Array for the location = -1
Rpt Period NOx Mass Calculated Accumulator Array for the location = -1
Return result B

Else if (***NOx Mass Derived Hourly Count*** > 1)
Rpt Period NOx Mass Reported Accumulator Array for the location = -1
Rpt Period NOx Mass Calculated Accumulator Array for the location = -1
Return result C

Else if (***NOx Mass Derived Hourly Count*** > 0 AND ***Current NOx Rate Method Code*** == "AE" AND ***Hourly Fuel Flow Count for Gas + Hourly Fuel Flow Count for Oil*** == 0)
Rpt Period NOx Mass Reported Accumulator Array for the location = -1
Rpt Period NOx Mass Calculated Accumulator Array for the location = -1
Return result E

Else if (***NOx Mass Derived Hourly Count*** == 1)

Current NOX Mass Derived Hourly Record = DerivedHourlyValueData rec matching with param NOX or NOXM where ***Current Date*** = DerivedHourlyValueData.Date and ***Current Hour*** = DerivedHourlyValueData.Hour

If (***LME HI Method*** is not null)
If (***NOx Mass Monitor Method Code*** == "LME")
if (***Current NOX Mass Derived Hourly Record***.ParameterCode == "NOXM")
NOXM Derived Checks Needed = true
else
Rpt Period NOx Mass Reported Accumulator Array for the location = -1
Rpt Period NOx Mass Calculated Accumulator Array for the location = -1
retunr result F
else
Rpt Period NOx Mass Reported Accumulator Array for the location = -1

Rpt Period NOx Mass Calculated Accumulator Array for the location = -1

else

if (***Current NOX Mass Derived Hourly Record***.ParameterCode == "NOXM")
Rpt Period NOx Mass Reported Accumulator Array for the location = -1
Rpt Period NOx Mass Calculated Accumulator Array for the location = -1
retunr result F

else
NOx Mass Derived Checks Needed = true

If (***NOx Mass Monitor Method Code*** in set {AMS, CEMNOXR})

If (***NOx Rate Derived Hourly Count*** > 0)
NOx Mass Monitor Method Code = "NOXR"

Else if (***NOx Mass Monitor Method Code*** == "CEMNOXR")
NOx Mass Monitor Method Code == "CEM"

Else if (***Current NOx Mass Derived Hourly Record***.FormulaIdentifier is not null)

NOX Formula Record = MonitorFormulaData record where
MonitorFormulaData.FormulaID = ***Current NOx Mass Derived Hourly Record.***FormulaIdentifier

If (NOX Formula Record is not null)

If (NOX Formula Record.ParameterCode == "NOX" AND NOX Formula Record.EquationCode in set {F-26A,F-26B})
NOx Mass Monitor Method Code = "CEM"

Apportionment NOX Method Array at this location = ***NOx Mass Monitor Method Code***

else
If ***NOx Mass Derived Hourly Count*** > 0
Return result D

Results:

Result	Response	Severity
A	You did not report a DHV record for NOX (or NOXM) for the hour.	Critical Error Level 1
B	You reported a DHV record for NOX (or NOXM), but you did not report an active NOX (or NOXM) method record in your monitoring plan for the hour.	Critical Error Level 1
C	You reported more than one DHV record for NOX (or NOXM) for the hour.	Critical Error Level 1
D	You reported a DHV record for NOX (or NOXM), but this is not appropriate for a non-operating hour.	Critical Error Level 1
E	You reported a DHV record for [param], but you did not report any Hourly Fuel Flow records at the location.	Critical Error Level 1
F	The ParameterCode reported in the DHV record does not match the ParameterCode in the Method record in your monitoring plan used to determine [eparam].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-22

Check Name: Verify Single CO2 Mass Derived Hourly Value Record

Related Former Checks:

Applicability: CEM Check

Description: Verifies that exactly ONE Derived Hourly Value record exists for the current hour associated with CO2 Mass

Specifications:

If (***Derived Hourly Checks Needed*** == true)
CO2 Mass Derived Checks Needed = false
Current CO2 Mass Derived Hourly Record = null
CO2M Derived Checks Needed = false

CO2 Mass Derived Hourly Count = count of DerivedHourlyValueData records with ParameterCode beginning with "CO2" where
Current Date = DerivedHourlyValueData.Date and
Current Hour = DerivedHourlyValueData.Hour

If ***Current Hourly Op Record***.OperatingTime > 0
If (***CO2 Mass Derived Hourly Count*** == 0 AND ***CO2 Method Code*** is not null AND ***CO2 Method Code*** <> "FSA")
If (***CO2 Method Code*** == "AD")
If (***Hourly Fuel Flow Count for Gas + Hourly Fuel Flow Count for Oil*** > 0)
return result A
else
Return result A
Else if (***CO2 Mass Derived Hourly Count*** > 0 AND (***CO2 Method Code*** is null OR ***CO2 Method Code*** == "FSA"))
Rpt Period CO2 Mass Reported Accumulator Array for the location = -1
Rpt Period CO2 Mass Calculated Accumulator Array for the location = -1
return result B
Else if (***CO2 Mass Derived Hourly Count*** > 1)
Rpt Period CO2 Mass Reported Accumulator Array for the location = -1
Rpt Period CO2 Mass Calculated Accumulator Array for the location = -1
Return result C
Else if (***CO2 Mass Derived Hourly Count*** > 0 AND ***CO2 Method Code*** == "AD" AND ***Hourly Fuel Flow Count for Gas + Hourly Fuel Flow Count for Oil*** == 0)
Rpt Period CO2 Mass Reported Accumulator Array for the location = -1
Rpt Period CO2 Mass Calculated Accumulator Array for the location = -1
Return result E
Else if (***CO2 Mass Derived Hourly Count*** == 1)

Current CO2 Mass Derived Hourly Record = DerivedHourlyValueData rec matching with param CO2 or CO2M where ***Current Date*** = DerivedHourlyValueData.Date and ***Current Hour*** = DerivedHourlyValueData.Hour

If (***LME HI Method*** is not null)
If (***CO2 Method Code*** == "LME")

if (***Current CO2 Mass Derived Hourly Record***.ParameterCode == "CO2M")
CO2M Derived Checks Needed = true
else
Rpt Period CO2 Mass Reported Accumulator Array for the location = -1
Rpt Period CO2 Mass Calculated Accumulator Array for the location = -1
return result F
else
Rpt Period CO2 Mass Reported Accumulator Array for the location = -1
Rpt Period CO2 Mass Calculated Accumulator Array for the location = -1
else
if (***Current CO2 Mass Derived Hourly Record***.ParameterCode == "CO2M")

Rpt Period CO2 Mass Reported Accumulator Array for the location = -1
Rpt Period CO2 Mass Calculated Accumulator Array for the location = -1
return result F

else
CO2 Mass Derived Checks Needed = true

If (***CO2 Method Code*** == "AMS")

if (***Current CO2 Mass Derived Hourly Record***.FormulaIdentifier is not null)

CO2 Formula Record = MonitorFormulaData record where
MonitorFormulaData.FormulaID = ***Current CO2 Mass Derived Hourly Record.***FormulaIdentifier

If (CO2 Formula Record is not null)

If (CO2 Formula Record.ParameterCode == "CO2" AND CO2 Formula Record.EquationCode in set {F-2,F-11})
CO2 Method Code == "CEM"
CO2 CEM Method Active For Hour = true

Else
If ***CO2 Mass Derived Hourly Count*** > 0
Return result D

Results:

Result	Response	Severity
A	You did not report a DHV record for CO2 (or CO2M) for the hour.	Critical Error Level 1
B	You reported a DHV record for CO2 (or CO2M), but you did not report an active CO2 (or CO2M) method record in your monitoring plan for the hour.	Critical Error Level 1
C	You reported more than one DHV records for CO2 (or CO2M) for the hour.	Critical Error Level 1
D	You reported a DHV record for CO2 (or CO2M), but this is not appropriate for a non-operating hour.	Critical Error Level 1
E	You reported a DHV record for [param], but you did not report any Hourly Fuel Flow records at the location.	Critical Error Level 1
F	The ParameterCode reported in the DHV record does not match the ParameterCode in the Method record in your monitoring plan used to determine [eparam].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-23

Check Name: Verify CO2 Conc Derived and Monitor Hourly Data Record

Related Former Checks:

Applicability: CEM Check

Description: This check scans the DerivedHourlyValueData and MonitorHourlyValueData records to ensure that a single record containing CO2 concentration values is reported for the current hour

Specifications:

Current CO2 Conc Derived Hourly Record = null
Current CO2 Conc Monitor Hourly Record = null
Current CO2 Conc Missing Data Monitor Hourly Record = null
CO2 Conc Derived Checks Needed = false
CO2 Conc Monitor Checks Needed = false
O2 Dry Needed to Support CO2 Calculation = false
O2 Wet Needed to Support CO2 Calculation = false

CO2C Has Measured DHV MODC = null

CO2 Conc Derived Hourly Count = count of DerivedHourlyValueData records with ParameterCode = "CO2C" where
Current Date = DerivedHourlyValueData.Date and
Current Hour = DerivedHourlyValueData.Hour

CO2 Conc Monitor Hourly Count = count of MonitorHourlyValueData records with ParameterCode = "CO2C" where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

Total CO2 Conc Records = ***CO2 Conc Derived Hourly Count*** + ***CO2 Conc Monitor Hourly Count***

If (***Current Hourly Op Record***.OperatingTime > 0)

If ((***CO2 Conc Checks Needed for Heat Input*** == true) OR (***CO2 Diluent Checks Needed for NOx Rate Calc*** == true) OR (***CO2 Diluent Needed for MATS*** == true) OR (***CO2 Conc Checks Needed for CO2 Mass Calc*** == true))

If ((***CO2 Conc Monitor Hourly Count*** == 0) AND ((***CO2 Conc Checks Needed for Heat Input*** == true) OR (***CO2 Diluent Checks Needed for NOx Rate Calc*** == true) OR (***CO2 Diluent Needed for MATS*** == true)))

If (***CO2 Conc Checks Needed for Heat Input*** == true) OR
(***CO2 Diluent Checks Needed for NOx Rate Calc*** == true AND ***NOXR Has Measured DHV MODC*** == true) OR
(***CO2 Diluent Needed for MATS*** == true AND ***CO2 Diluent Needed for MATS Calculation*** == true)
return result B
Else
return result F

else if (*Total CO2 Conc Records* == 0)
return result A

else if ((***CO2 Conc Monitor Hourly Count*** == 2) AND (***CO2 Conc Derived Hourly Count*** == 0) AND ((***CO2 Diluent Checks Needed for NOx Rate Calc*** == true) OR (***CO2 Diluent Needed for MATS*** == true)) AND ((***CO2 Conc Checks Needed for Heat Input*** == true) OR (***CO2 Conc Checks Needed for CO2 Mass Calc*** = true))

Current CO2 Conc Monitor Hourly Record = Find MonitorHourlyValueData records with ParameterCode = "CO2C" and MODCCode in set {01, 02, 03, 04, 53, 54} where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

Current CO2 Conc Missing Data Monitor Hourly Record = Find MonitorHourlyValueData records with ParameterCode = "CO2C" and MODCCode not in set {01, 02, 03, 04, 54} where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

If (***Current CO2 Conc Monitor Hourly Record*** is null OR ***Current CO2 Conc Missing Data Monitor Hourly Record*** is null)
return result C

else
CO2 Conc Monitor Checks Needed = true

else if (*Total CO2 Conc Records* > 1)
return result C

else if (***CO2 Conc Derived Hourly Count*** == 1)

CO2 Conc Derived Checks Needed = true
Current CO2 Conc Derived Hourly Record = matching DerivedHourlyValueData rec
CO2C Has Measured DHV MODC = (***Current CO2 Conc Derived Hourly Record***.MODCCode in set {01, 02, 03, 04, 05, 21, 53, 54}

if (***Current CO2 Conc Derived Hourly Record***.MODCCode in set {01, 02, 03, 04, 05, 21, 53, 54}

Fc Factor Needed = true
Fd Factor Needed = true

If (***Current CO2 Conc Derived Hourly Record***.Formula Id Key is not null)

CO2C Formula record = Find MonitoringFormulaData record where
MonitoringFormulaIDKey = ***Current CO2 Conc Derived Hourly Record***.Formula Id Key

If (*CO2C Formula record* is not null)

If (*CO2C Formula record*.ParameterCode == "CO2C")

If (CO2C Formula record.EquationCode == "F-14A")
O2 Dry Needed to Support CO2 Calculation = true

else if (CO2C Formula record.EquationCode == "F-14B")
O2 Wet Needed to Support CO2 Calculation = true
Moisture Needed = true

else if (***CO2 Conc Monitor Hourly Count*** == 1)
CO2 Conc Monitor Checks Needed = true
Current CO2 Conc Monitor Hourly Record = matching MonitorHourlyValueData rec

else
If (*Total CO2 Conc Records* > 0)
Return result D

else
If (*Total CO2 Conc Records* > 0)
Return result E

Results:

Result	Response	Severity
A	You did not report a MHV or DHV record for CO2C for the hour.	Critical Error Level 1
B	You did not report an MHV record for CO2C for the hour.	Critical Error Level 1
C	You reported more than one MHV and/or DHV records for CO2C for the hour.	Critical Error Level 1
D	You reported a MHV or DHV record for CO2C, but this record is not required to calculate emissions.	Non-Critical Error
E	You reported a MHV or DHV record for CO2C, but this is not appropriate for a non-operating hour.	Critical Error Level 1
F	You did not report an MHV record for CO2C for the hour.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-24

Check Name: Count Hourly Fuel Flow Records

Related Former Checks:

Applicability: General Check

Description: Counts the number of Hourly Fuel Flow Records for the current hour and checks for consistency with Appendix D and/or Appendix E Methods

Specifications:

If (***Derived Hourly Checks Needed*** == true)
Hourly Fuel Flow Count For Oil = 0
Hourly Fuel Flow Count For Gas = 0

Appendix D Method Active = ***Heat Input App D Method Active For Hour*** OR
CO2 App D Method Active For Hour OR
SO2 App D Method Active For Hour

Hourly Fuel Flow List = set of all DerivedHourlyValueData records
Current Date = DerivedHourlyValueData.Date and
Current Hour = DerivedHourlyValueData.Hour

For each record (*Current Hourly Fuel Flow Record*) in *Hourly Fuel Flow List*

Cur Fuel Code = *Current Hourly Fuel Flow Record*.FuelCode
if (*Cur Fuel Code* is null)
return result D
Current Fuel Group = select FuelGroupCode from FuelCode Table where FuelCode = *Cur Fuel Code*
if (*Current Fuel Group* is null)
return result D
else If Current Fuel Group == "GAS"
Add 1 to ***Hourly Fuel Flow Count For Gas***
else if Current Fuel Group == "OIL"
Add 1 to ***Hourly Fuel Flow Count For Oil***

Hourly Fuel Flow Count = ***Hourly Fuel Flow Count For Gas*** + ***Hourly Fuel Flow Count For Oil***

If (***Current Hourly Op Record***.LocationName begins with "CP")
CP Fuel Count = ***CP Fuel Count*** + ***Hourly Fuel Flow Count***

If (***Current Hourly Op Record***.OperatingTime== 0)
If (***Hourly Fuel Flow Count*** > 0)
Return result A
else
if (***Appendix D Method Active*** = true AND ***Hourly Fuel Flow Count*** == 0 AND ***MP Pipe Config for Hourly Checks*** is null)
Return result B
else if (***Appendix D Method Active*** = false AND ***Hourly Fuel Flow Count*** > 0)
return result C

Results:

Result	Response	Severity
A	You reported an HFF record, but this is not appropriate for a non-operating hour.	Critical Error Level 1
B	You did not report an HFF record for the hour.	Critical Error Level 1
C	You reported a HFF record, but you did not report an active AD or AE method record in your monitoring plan for the hour.	Critical Error Level 1
D	The FuelCode reported in the HFF record is missing or invalid.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-30

Check Name: Determine Load Based Status of unit

Related Former Checks:

Applicability: General Check

Description: Determines whether current entity is load based

Specifications:

Unit is Load Based = false

Location Name = ***Current Monitor Plan Location Record***.LOCATION_NAME

if the Location Name begins with "CS" or "CP" or "MS" or "MP"

Locate all Unit Stack Configuration records where the stack/pipe location is the monitoring location, the BeginDate is on or before the Current Date, and the EndDate is null or is on or after the Current Date.

If the NonLoadBasedIndicator in all of the retrieved records is equal to 1,

Unit is Load Based = false

else

Unit is Load Based = true

else // current location is a unit

if the NonLoadBasedIndicator field for the unit = 1

Unit is Load Based = false

else

Unit is Load Based = true

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-32

Check Name: Perform Load Checks for Operating Hour

Related Former Checks:

Applicability: General Check

Description: Checks to see that Load is populated correctly for operating hours. Also checks Units of Measure Code for Load

Specifications:

CurrentMaximumLoadValue = null

if (***Current Hourly Op Record*** is not null)

Apportionment OpTime Array for this location = ***Current Hourly Op Record***.OperatingTime
Apportionment Load Array for this Location = ***Current Hourly Op Record***.HourlyLoad

if (***Unit is Load Based*** == true and ***Current Hourly Op Record***.OperatingTime > 0.0)

if (***Current Hourly Op Record***.HourLoad is null OR ***Current Hourly Op Record***.HourLoad < 0)
if (***Current Entity Type*** = "Unit")
Unit LoadTimesOpTime Accumulator = -1
else if ***(Current Entity Type*** in set {CP, MP})
Pipe LoadTimesOpTime Accumulator = -1
else
Stack LoadTimesOpTime Accumulator = -1
return result A
else
if (***MP Stack Config for Hourly Checks*** == "MS" AND ***Current Entity Type*** == "Unit")
MP Unit Load = ***Current Hourly Op Record***.HourLoad

if (***Current Entity Type*** = "Unit")
if (***Unit LoadTimesOpTime Accumulator*** >= 0)
Unit LoadTimesOpTime Accumulator = ***Unit LoadTimesOpTime Accumulator*** +
(***Current Hourly Op Record***.HourLoad * ***Current Hourly Op Record***.OperatingTime)

else if ***(Current Entity Type*** in set {CP, MP})
if (***Pipe LoadTimesOpTime Accumulator*** >= 0)
Pipe LoadTimesOpTime Accumulator = ***Pipe LoadTimesOpTime Accumulator*** +
(***Current Hourly Op Record***.HourLoad * ***Current Hourly Op Record***.OperatingTime)

else
if (***Stack LoadTimesOpTime Accumulator*** >= 0)
Stack LoadTimesOpTime Accumulator = ***Stack LoadTimesOpTime Accumulator*** +
(***Current Hourly Op Record***.HourLoad * ***Current Hourly Op Record***.OperatingTime)

if ***Current Hourly Op Record.***LoadUnitsOfMeasureCode not in {"MW","KLBHR", "MMBTUHR"}
MP Load UOM = "INVALID"
return result B

else if (***MP Load UOM*** is not null AND ***MP Load UOM*** <> "INVALID" AND ***MP Load UOM*** <> ***Current Hourly Op Record.***LoadUnitsOfMeasureCode)
MP Load UOM = "INVALID"
return result C
else
if (***MP Load UOM*** is null)
MP Load UOM = ***Current Hourly Op Record.***LoadUnitsOfMeasureCode

Locate the ***MonitorLoadRecordsByHourandLocation*** record for the hour and location.

If (only one record is found AND ***MonitorLoadRecordByHourandLocation.***MaximumLoadValue is greater than 0),
    If (***Current Hourly Op Record.***LoadUnitsOfMeasureCode ==
    MonitorLoadRecordByHourandLocation.MaximumLoadUnitsOfMeasureCode)
        If (***Current Hourly Op Record***.HourLoad is greater than
        MonitorLoadRecordByHourandLocation.MaximumLoadValue)
            If (***Current Hourly Op Record***.HourLoad is greater than 1.25 *
            MonitorLoadRecordByHourandLocation.MaximumLoadValue)
                return result L
            else
                return result H
        Else
            CurrentMaximumLoadValue =
            MonitorLoadRecordByHourandLocation.MaximumLoadValue
    else
        return result I
else
    return result J

else if (***Current Hourly Op Record***.OperatingTime == 0.0)
    if (***Current Hourly Op Record***.HourLoad is not null)
        return result D
    if ***Current Hourly Op Record.***LoadUnitsOfMeasureCode is not null
        return result E

else if (***Unit is Load Based*** == false)
    if (***Current Hourly Op Record***.HourLoad is not null)
        return result F
    if ***Current Hourly Op Record.***LoadUnitsOfMeasureCode is not null
        return result G

Results:

Result	Response	Severity
A	The HourLoad reported in the Hourly Operating record is invalid. The value must be greater than or equal to 0.	Critical Error Level 1
B	The LoadUnitsOfMeasureCode reported in the Hourly Operating record is invalid.	Critical Error Level 1
C	You did not report the same LoadUnitsOfMeasureCode for all locations in the configuration.	Critical Error Level 1
D	You reported HourLoad in the Hourly Operating record. This field should be blank for a non-operating hour.	Critical Error Level 1
E	You reported a LoadUnitsOfMeasureCode in the Hourly Operating record. This field should be blank for a non-operating hour.	Non-Critical Error
F	You reported HourLoad in the Hourly Operating record. This field should be blank for a non-load-based unit.	Critical Error Level 1
G	You reported a LoadUnitsOfMeasureCode in the Hourly Operating record. This field should be blank for a non-load-based unit.	Critical Error Level 1
H	Warning: The HourLoad reported in the Hourly Operating Data record is higher than the MaximumLoadValue in the Monitoring Load record reported in your monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. You should investigate the cause of these exceedances and determine whether an adjustment to the MaximumLoadValue in your monitoring plan is necessary.	Informational Message
I	The [fieldname] does not correspond to the MaximumLoadUnitsOfMeasure reported in the monitoring plan.	Critical Error Level 2
J	You did not have one and only one valid Monitor Load record that was active during the hour.	Critical Error Level 1
K	The LoadRange or CommonStackLoadRange reported in the Hourly Operating record is inconsistent with the HourLoad. When no load is generated, the load range should be less than 2.	Informational Message
L	You reported an HourLoad in the Hourly Operating Data record that is 125% or greater than the MaximumLoadValue in the Monitoring Load record reported in your monitoring plan.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-33

Check Name: Check reported Fuel Code for Operating Hour

Related Former Checks:

Applicability: General Check

Description: Where applicable, ensures that the fuel code is valid

Specifications:

if (***Current Hourly Op Record*** is not null)

Fuel Code Validation Needed = false

If (***NOx Rate Fuel Specific Missing Data*** == true OR ***NOx Mass Fuel Specific Missing Data*** == true OR ***SO2 Fuel Specific Missing Data*** == true OR ***CO2 Fuel Specific Missing Data*** == true OR ***Heat Input Fuel Specific Missing Data*** == true OR ***H2O Fuel Specific Missing Data*** == true)
Fuel Code Validation Needed = true

else
if (***Current SO2 Monitor Hourly Record*** is not null AND ***SO2 Bypass Code*** == "BYMAXFS")
if (***Current SO2 Monitor Hourly Record***.MODCCode == 23)
Fuel Code Validation Needed = true

if (***Current NOx Conc Monitor Hourly Record*** is not null AND ***NOx Mass Bypass Code*** == "BYMAXFS")
if (***Current NOx Conc Monitor Hourly Record***.MODCCode in set {23,24})
Fuel Code Validation Needed = true

if (***Current NOx Rate Derived Hourly Record*** is not null AND ***NOx Rate Bypass Code*** == "BYMAXFS")
if (***Current NOx Rate Derived Hourly Record***.MODCCode in set {23,24})
Fuel Code Validation Needed = true

if (*Fuel Code Validation Needed* == true)
if (***Current Hourly Op Record***.FuelCode is null)
If (***Current Hourly Op Record***.OperatingTime is greater than 0)
return result A
else
Current Hourly Fuel Group Code = FuelGroupCode from FuelCode table entry where
FuelCode = ***Current Hourly Op Record***.FuelCode
if (***Current Hourly Op Record***.FuelCode = "NFS" OR
(*Current Hourly Fuel Group Code* == "COAL" AND ***Current Hourly Op Record***.FuelCode <> "C"))
return result B

else if (***Current Hourly Op Record***.FuelCode is not null)
if (***SO2 Bypass Code*** <> "BYMAXFS" AND ***NOx Rate Bypass Code*** <> "BYMAXFS" AND ***NOx Mass Bypass Code*** <> "BYMAXFS")
return result C

Results:

Result	Response	Severity
A	You did not report a FuelCode in the Hourly Operating record.	Critical Error Level 1
B	The FuelCode reported Hourly Operating record is invalid.	Critical Error Level 1
C	You reported a FuelCode in the Hourly Operating record. This value should only be reported if you use fuel-specific missing data or have an unmonitored bypass stack that reports emissions based on fuel-specific maximum values.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-34

Check Name: Validate Reported FC Factor

Related Former Checks:

Applicability: General Check

Description: Uses cross-check value to ensure that FC Factor reported in Hourly Operating Data is within acceptable range

Validation Tables:

F-Factor Range Checks (Cross Check Table)

Specifications:

if ***FC Factor Needed*** = true
Valid FC Factor Exists = false

if (***Current Hourly Op Record***.FcFactor is null)
return result A

else if (***Current Hourly Op Record***.FcFactor <= 0)
return result A

else
Valid FC Factor Exists = true
if (***Special Fuel Burned*** = false)

FC Factor Minimum = Lookup Lower from Cross-Check Table "F-Factor Range Checks" where Factor = "FC"
FC Factor Maximum = Lookup Upper from Cross-Check Table "F-Factor Range Checks" where Factor = "FC"
if (***Current Hourly Op Record***.FcFactor > ***FC Factor Maximum*** OR ***Current Hourly Op Record***.FcFactor < ***FC Factor Minimum***)
return result B

Results:

Result	Response	Severity
A	The [FNAME] reported in the Hourly Operating record is missing or invalid.	Critical Error Level 1
B	The [FNAME] reported in the Hourly Operating record is outside of the expected range from [MIN] to [MAX].	Critical Error Level 2

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-35

Check Name: Validate Reported FD Factor

Related Former Checks:

Applicability:

Description: Uses cross-check value to ensure that FD Factor reported in Hourly Operating Data is within acceptable range

Validation Tables:

F-Factor Range Checks (Cross Check Table)

Specifications:

if ***FD Factor Needed*** = true
Valid FD Factor Exists = false

if (***Current Hourly Op Record***.FdFactor is null)
return result A

else if (***Current Hourly Op Record***.FdFactor <= 0)
return result A

else
Valid FD Factor Exists = true

if (***Special Fuel Burned*** = false)
FD Factor Minimum = Lookup Lower from Cross-Check Table "F-Factor Range Checks" where Factor = "FD"
FD Factor Maximum = Lookup Upper from Cross-Check Table "F-Factor Range Checks" where Factor = "FD"

if (***Current Hourly Op Record***.FdFactor > ***FD Factor Maximum*** OR ***Current Hourly Op Record***.FdFactor < ***FD Factor Minimum***)
return result B

Results:

Result	Response	Severity
A	The [FNAME] reported in the Hourly Operating record is missing or invalid.	Critical Error Level 1
B	The [FNAME] reported in the Hourly Operating record is outside of the expected range from [MIN] to [MAX].	Critical Error Level 2

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-36

Check Name: Validate Reported FW Factor

Related Former Checks:

Applicability: General Check

Description: Uses cross-check value to ensure that FW Factor reported in Hourly Operating Data is within acceptable range

Validation Tables:

F-Factor Range Checks (Cross Check Table)

Specifications:

if ***FW Factor Needed*** = true
Valid FW Factor Exists = false

if (***Current Hourly Op Record***.FwFactor is null)
return result A

else if (***Current Hourly Op Record***.FwFactor <= 0)
return result A

else
Valid FW Factor Exists = true

if ***(Special Fuel Burned*** = false)
FW Factor Minimum = Lookup Lower from Cross-Check Table "F-Factor Range Checks" where Factor = "FW"
FWFactor Maximum = Lookup Upper from Cross-Check Table "F-Factor Range Checks" where Factor = "FW"

if (***Current Hourly Op Record***.FwFactor > ***FW Factor Maximum*** OR ***Current Hourly Op Record***.FwFactor < ***FW Factor Minimum***)
return result B

Results:

Result	Response	Severity
A	The [FNAME] reported in the Hourly Operating record is missing or invalid.	Critical Error Level 1
B	The [FNAME] reported in the Hourly Operating record is outside of the expected range from [MIN] to [MAX].	Critical Error Level 2

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-37

Check Name: Verify Single Heat Input Derived Hourly Record

Related Former Checks:

Applicability: CEM Check

Description: Verify that a single Derived Hourly record exists for Heat Input for the current location and hour

Specifications:

Current Heat Input Derived Hourly Record= null
Heat Input Derived Checks Needed = false
HIT Derived Checks Needed = false
Heat Input Derived Hourly Count = # of DerivedHourlyValueData record with parameter beginning with "HI" where
Current Date = DerivedHourlyValueData.Date and
Current Hour = DerivedHourlyValueData.Hour

If ***Current Hourly Op Record***.OperatingTime > 0
If (Heat Input Derived Hourly Count == 0)
If (***Heat Input Method Code*** is not null)
If (***Heat Input Method Code*** not in set {EXP, LTFF})
return result A
else if (***Heat Input Method Code*** == "LTFF" AND ***Current Entity Type*** == "Unit")
return result A
Else if (Heat Input Derived Hourly Count > 0 AND
(***Heat Input Method Code*** is null OR
Heat Input Method Code == "EXP" OR
(***Heat Input Method Code*** == "LTFF" AND ***Current Entity Type*** == "CP"))
Rpt Period HI Reported Accumulator Array for the location = -1
Rpt Period HI Calculated Accumulator Array for the location = -1
return result B
Else if (Heat Input Derived Hourly Count >1)
Rpt Period HI Reported Accumulator Array for the location = -1
Rpt Period HI Calculated Accumulator Array for the location = -1
return result C
Else
Current Heat Input Derived Hourly Record= DerivedHourlyValueData record with parameter "HI" or "HIT" where
Current Date = DerivedHourlyValueData.Date and
Current Hour = DerivedHourlyValueData.Hour

If (***LME HI Method*** is not null)

if (***Current Heat Input Derived Hourly Record***.ParameterCode == "HIT")
HIT Derived Checks Needed = true
else
Rpt Period HI Reported Accumulator Array for the location = -1
Rpt Period HI Calculated Accumulator Array for the location = -1
return result E
else
if (***Current Heat Input Derived Hourly Record***.ParameterCode == "HIT")
Rpt Period HI Reported Accumulator Array for the location = -1
Rpt Period HI Calculated Accumulator Array for the location = -1
return result E

else
Heat Input Derived Checks Needed = true

If (***Heat Input Method Code*** == "AMS")

if (***Current Heat Input Derived Hourly Record***.FormulaIdentifier is not null)

HI Formula Record = MonitorFormulaData record where
MonitorFormulaData.FormulaID = ***Current Heat Input Derived Hourly Record.***FormulaIdentifier

If (HI Formula Record is not null)

If (HI Formula Record.ParameterCode == "HI" AND HI Formula Record.EquationCode in set {F-15,F-16,F-17,F-18})
Heat Input Method Code == "CEM"
Heat Input CEM Method Active For Hour == true

else
If Heat Input Derived Hourly Count > 0
return result D

Results:

Result	Response	Severity
A	You did not report a DHV record for HI (or HIT) for the hour. If you have entered LME data via the LME Emissions Data Utility, this error indicates that you have not yet generated your quarterly emissions data. You must do this by clicking on the Generate Emissions Data link on the LME Emissions Data Utility submenu.	Critical Error Level 1
B	You reported a DHV record for HI (or HIT), but, according to the monitoring methods in your monitoring plan, you should not report hourly heat input at this location.	Critical Error Level 1
C	You reported more than one DHV record for HI (or HIT) for the hour.	Critical Error Level 1
D	You reported a DHV record for HI (or HIT), but this is not appropriate for a non-operating hour.	Critical Error Level 1
E	The ParameterCode reported in the DHV record does not match the ParameterCode in the Method record in your monitoring plan used to determine [eparam].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-38

Check Name: Determine Fuel Type

Related Former Checks:

Applicability: General Check

Description:

Specifications:

If (***Derived Hourly Checks Needed*** == true)

If ***Current Hourly Op Record***.FcFactor is not null OR ***Current Hourly Op Record***.FdFactor is not null OR ***Current Hourly Op Record***.FwFactor is not null

If (***Hourly Fuel Flow Count For Oil*** + ***Hourly Fuel Flow Count For Gas*** == 0)

If ***(Current Hourly Op Record***.FuelCd is null OR ***Current Hourly Op Record***.FuelCd == "MIX")

Count all active UnitFuel records for the location where
FuelCd in set {OOL, PRG, PRS, OGS}

If count > 0
Special Fuel Burned = true

else if ***Current Hourly Op Record***.UnitFuelCd in set {OOL, PRG, PRS, OGS}
Special Fuel Burned = true

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-39

Check Name: Verify Single H2O Conc Derived or Monitor Hourly Data Record

Related Former Checks:

Applicability: CEM Check

Description: This check scans the DerivedHourlyValueData and MonitorHourlyValueData records to ensure that a single record containing H2O concentration values is reported for the current hour

Specifications:

H2O Monitor Hourly Checks Needed = false
H2O Derived Hourly Checks Needed = false
Current H2O Monitor Hourly Record = null
Current H2O Derived Hourly Record = null
O2 Wet Checks Needed for H2O= false
O2 Dry Checks Needed for H2O= false
H2O Has Measured DHV MODC=null

If ***Current Hourly Op Record***.OperatingTime > 0.00

If (**Moisture Needed** == true)

If ***H2O Monitor Hourly Count + H2O Derived Hourly Count*** == 0

If (***H2O Method Code*** == "MWD")
return result A

Else if (***H2O Method Code*** <> "MDF")
return result B

Else if ***H2O Default Max Value*** is not null
return result C

Else if ***(H2O Derived Hourly Count*** > 1)
return result D

Else if ***(H2O Monitor Hourly Count*** > 1)
return result E

Else if (***H2O Derived Hourly Count*** == 1 AND ***H2O Method Code*** in set {MTB, MMS})
return result F

Else if (***H2O Monitor Hourly Count*** == 1 AND ***H2O Method Code*** in set {MWD, MDF})
return result G

Else if ***(H2O Monitor Hourly Count*** == 1)

Current H2O Monitor Hourly Record = MonitorHourlyValueData record matching with ParameterCode = "H2O" where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

H2O Monitor Hourly Checks Needed = true

Else if (***H2O Derived Hourly Count*** == 1)

Current H2O Derived Hourly Record = DerivedHourlyValueData rec matching where
DerivedHourlyValueData.ParameterCode = "H2O" AND

DerivedHourlyValueData.Date = ***Current Date*** AND
DerivedHourlyValueData.Hour = ***Current Hour***

H2O Has Measured DHV MODC = (***Current H2O Derived Hourly Record***.ModcCode in set {01, 02, 03, 04, 05, 21, 53, 54})

H2O Derived Hourly Checks Needed = true

if (***Current H2O Derived Hourly Record***.ModcCode in set {01, 02, 03, 04, 05, 21, 53, 54} AND ***Current H2O Derived Hourly Record.***FormulaIdentifier is not null)

H2O Formula Record = MonitorFormulaData record where
MonitorFormulaData.FormulaID = ***Current H2O Derived Hourly Record.***FormulaIdentifier

If (*H2O Formula Record* is not null)

If (*H2O Formula Record*.ParameterCode == "H2O" AND H2O Formula Record.EquationCode in set {F-31, M-1K})
O2 Wet Checks Needed for H2O = true
O2 Dry Checks Needed for H2O = true

else
If ***H2O Monitor Hourly Count + H2O Derived Hourly Count*** > 0
return result I

Results:

Result	Response	Severity
A	You did not report a DHV record for H2O for the hour.	Critical Error Level 1
B	You did not report an MHV record for H2O for the hour.	Critical Error Level 1
C	You did not report a DHV record for H2O for the hour. You must report this record if you have multiple H2O default values for different fuels.	Critical Error Level 1
D	You reported more than one DHV record for H2O for the hour.	Critical Error Level 1
E	You reported more than one MHV record for H2O for the hour.	Critical Error Level 1
F	You reported a DHV record for H2O, but the H2O MethodCode is not "MWD" or "MDF".	Critical Error Level 1
G	You reported a MHV record for H2O, but the H2O MethodCode is not "MTB" or "MMS".	Critical Error Level 1
H	You reported a DHV and/or MHV record for H2O, but this record is not required to calculate emissions.	Informational Message
I	You reported a DHV and/or MHV record for H2O, but this is not appropriate for a non-operating hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-40

Check Name: Verify Single O2 Dry Monitor Hourly Value Record

Related Former Checks:

Applicability: CEM Check

Description: This check scans the MonitorHourlyValueData records to ensure that a single O2 Wet exists for the current hour.

Note that Result A and G indicate that a dry moisture basis is required because both a wet and dry O2C are needed. Otherwise a null moisture basis can be used.

Specifications:

Current O2 Dry Monitor Hourly Record = null
Current O2 Dry Missing Data Monitor Hourly Record = null
O2 Dry Monitor Hourly Checks Needed = false

If ***Current Hourly Op Record***.OperatingTime > 0.00

If ((***O2 Dry Checks Needed for Heat Input*** == true) OR (***O2 Dry Checks Needed for NOx Rate Calc*** == true) OR (***O2 Dry Checks Needed To Support CO2 Calculation*** == true) OR (***O2 Dry Checks Needed for H2O*** == true) OR (***O2 Dry Needed for MATS*** == true))

If ((***O2 Dry Monitor Hourly Count*** == 0) AND ((***O2 Wet Checks Needed for Heat Input*** == true) OR (***O2 Wet Checks Needed for NOx Rate Calc*** == true) OR (***O2 Wet Checks Needed To Support CO2 Calculation*** == true) OR (***O2 Wet Checks Needed for H2O*** == true) OR (***O2 Wet Needed for MATS*** == true)))

If (***O2 Wet Checks Needed for Heat Input*** == true) OR
(***O2 Wet Checks Needed for NOx Rate Calc*** == true AND ***NOXR Has Measured DHV MODC*** == true)
OR
(***O2 Wet Checks Needed To Support CO2 Calculation*** == true AND ***CO2C Has Measured DHV MODC*** == true) OR
(***O2 Wet Checks Needed for H2O*** == true AND ***H2O Has Measured DHV MODC*** == true) OR
(***O2 Wet Needed for MATS*** == true AND ***O2 Wet Needed for MATS Calculation*** == true)
return result A
Else
return result G

Else if (***O2 Dry Monitor Hourly Count + O2 Null Monitor Hourly Count*** == 0)

If (***O2 Dry Checks Needed for Heat Input*** == true) OR
(***O2 Dry Checks Needed for NOx Rate Calc*** == true AND ***NOXR Has Measured DHV MODC*** == true)
OR
(***O2 Dry Checks Needed To Support CO2 Calculation*** == true AND ***CO2C Has Measured DHV MODC*** == true) OR
(***O2 Dry Checks Needed for H2O*** == true AND ***H2O Has Measured DHV MODC*** == true) OR
(***O2 Dry Needed for MATS*** == true AND ***O2 Dry Needed for MATS Calculation*** == true)
return result B
Else
return result H

Else if ((***O2 Dry Monitor Hourly Count + O2 Null Monitor Hourly Count*** > 2) OR (***O2 Dry Monitor Hourly Count + O2 Null Monitor Hourly Count*** == 2 AND ***O2 Wet Monitor Hourly Count + O2 Null Monitor Hourly Count*** == 2))
Return result C

Else if (***O2 Dry Monitor Hourly Count + O2 Null Monitor Hourly Count*** == 2)

If (***O2 Dry Checks Needed for Heat Input*** == true AND (***O2 Dry Checks Needed for NOx Rate Calc*** == true OR

O2 Dry Checks Needed for H2O == true) OR (***O2 Dry Needed for MATS*** == true))

Current O2 Dry Monitor Hourly Record = Find MonitorHourlyValueData records with ParameterCode = "O2C" AND (MoistureBasis = "D" OR MoistureBasis is null) and MODCCode in set {01, 02, 03, 04, 53, 54} where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

Current O2 Dry Missing Data Monitor Hourly Record = Find MonitorHourlyValueData records with ParameterCode = "O2C" AND (MoistureBasis = "D" OR MoistureBasis is null) and MODCCode not in set {01, 02, 03, 04, 54} where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

If (***Current O2 Dry Monitor Hourly Record*** is null OR ***Current O2 Dry Missing Data Monitor Hourly Record*** is null)
return result C

else
O2 Dry Monitor Hourly Checks Needed = true

else
return result C

Else if (***O2 Dry Monitor Hourly Count*** == 1)

O2 Dry Monitor Hourly Checks Needed = true
Current O2 Dry Monitor Hourly Record = MonitorHourlyValueData record with ParameterCode = "O2C" AND MoistureBasis = "D" where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

Else if (***O2 Null Monitor Hourly Count*** == 1)

O2 Dry Monitor Hourly Checks Needed = true
Current O2 Dry Monitor Hourly Record = ***Current O2 Null Monitor Hourly Record***

else
If (***O2 Dry Monitor Hourly Count*** > 0)
Return result D

If ((***O2 Null Monitor Hourly Count*** > 0) AND (***O2 Wet Checks Needed for Heat Input*** == false) AND (***O2 Wet Checks Needed for NOx Rate Calc*** == false) AND (***O2 Wet Checks Needed To Support CO2 Calculation*** == false) AND (***O2 Wet Checks Needed for H2O*** == false) AND (***O2 Wet Needed for MATS*** == false))
return result E

else
If (***O2 Dry Monitor Hourly Count O2 + Null Monitor Hourly Count + O2 Wet Monitor Hourly Count*** > 0)
Return result F

Results:

Result	Response	Severity
A	You did not report a MHV record for O2C with a MoistureBasis of D for the hour.	Critical Error Level 1
B	You did not report a MHV record for O2C with a MoistureBasis of D (or blank) for the hour.	Critical Error Level 1
C	You reported too many MHV records for O2C with a MoistureBasis of D (or blank) for the hour.	Critical Error Level 1
D	You reported an MHV record for O2C with a MoistureBasis of [basis], but this record is not required to calculate emissions.	Non-Critical Error
E	You reported an MHV record for O2C with a blank MoistureBasis, but this record is not required to calculate emissions.	Non-Critical Error
F	You reported a MHV record for O2C, but this is not appropriate for a non-operating hour.	Critical Error Level 1
G	You did not report a MHV record for O2C with a MoistureBasis of D for the hour.	Critical Error Level 1
H	You did not report a MHV record for O2C with a MoistureBasis of D (or blank) for the hour.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-41

Check Name: Verify Single O2 Wet Monitor Hourly Value Record

Related Former Checks:

Applicability: CEM Check

Description: This check scans the MonitorHourlyValueData records to ensure that a single O2 Wet exists for the current hour. Otherwise a null moisture basis can be used.

Note that Result A and E indicate that a wet moisture basis is required because both a wet and dry O2C are needed.

Specifications:

Current O2 Wet Monitor Hourly Record = null
Current O2 Wet Missing Data Monitor Hourly Record = null
O2 Wet Monitor Hourly Checks Needed = false
If ***Current Hourly Op Record***.OperatingTime > 0.00

If ((***O2 Wet Checks Needed for Heat Input*** == true) OR (***O2 Wet Checks Needed for NOx Rate Calc*** == true) OR (***O2 Wet Checks Needed To Support CO2 Calculation*** == true) OR (***O2 Wet Checks Needed for H2O*** == true) OR (***O2 Wet Needed for MATS*** == true))

If ((***O2 Wet Monitor Hourly Count*** == 0) AND ((***O2 Dry Checks Needed for Heat Input*** == true) OR (***O2 Dry Checks Needed for NOx Rate Calc*** == true) OR (***O2 Dry Checks Needed To Support CO2 Calculation*** == true) OR (***O2 Dry Checks Needed for H2O*** == true) OR (***O2 Dry Needed for MATS*** == true)))

If (***O2 Dry Checks Needed for Heat Input*** == true) OR
(***O2 Dry Checks Needed for NOx Rate Calc*** == true AND ***NOXR Has Measured DHV MODC*** == true)
OR
(***O2 Dry Checks Needed To Support CO2 Calculation*** == true AND ***CO2C Has Measured DHV MODC*** == true) OR
(***O2 Dry Checks Needed for H2O*** == true AND ***H2O Has Measured DHV MODC*** == true) OR
(***O2 Dry Needed for MATS*** == true AND ***O2 Dry Needed for MATS Calculation*** == true)
return result A
Else
return result E

Else if (***O2 Wet Monitor Hourly Count + O2 Null Monitor Hourly Count*** == 0)

If (***O2 Wet Checks Needed for Heat Input*** == true) OR
(***O2 Wet Checks Needed for NOx Rate Calc*** == true AND ***NOXR Has Measured DHV MODC*** == true)
OR
(***O2 Wet Checks Needed To Support CO2 Calculation*** == true AND ***CO2C Has Measured DHV MODC*** == true) OR
(***O2 Wet Checks Needed for H2O*** == true AND ***H2O Has Measured DHV MODC*** == true) OR
(***O2 Wet Needed for MATS*** == true AND ***O2 Wet Needed for MATS Calculation*** == true)
return result B
Else
return result F

Return result B

Else if (***O2 Wet Monitor Hourly Count + O2 Null Monitor Hourly Count*** > 2)
Return result C

Else if (***O2 Wet Monitor Hourly Count*** + ***O2 Null Monitor Hourly Count*** == 2

If ((***O2 Wet Checks Needed for Heat Input*** == true) AND (***O2 Dry Monitor Hourly Count + O2 Null Monitor***

Hourly Count <> 2) AND ((***O2 Wet Checks Needed for NOx Rate Calc*** == true) OR (***O2 Wet Checks Needed for H2O*** == true) OR (***O2 Wet Needed for MATS*** == true)))

Current O2 Wet Monitor Hourly Record = Find MonitorHourlyValueData records with ParameterCode = "O2C" AND (MoistureBasis = "W" OR MoistureBasis is null) and MODCCode in set {01, 02, 03, 04, 53, 54} where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

Current O2 Wet Missing Data Monitor Hourly Record = Find MonitorHourlyValueData records with ParameterCode = "O2C" AND (MoistureBasis = "W" OR MoistureBasis is null) and MODCCode not in set {01, 02, 03, 04, 54} where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

If (***Current O2 Wet Monitor Hourly Record*** is null OR ***Current O2 Wet Missing Data Monitor Hourly Record*** is null)
return result C

else
O2 Wet Monitor Hourly Checks Needed = true

else
return result C

Else if (***O2 Wet Monitor Hourly Count*** == 1)

O2 Wet Monitor Hourly Checks Needed = true
Current O2 Wet Monitor Hourly Record = MonitorHourlyValueData record with ParameterCode = "O2C" AND MoistureBasis = "W" where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

Else if (***O2 Null Monitor Hourly Count*** == 1)

O2 Wet Monitor Hourly Checks Needed = true
Current O2 Wet Monitor Hourly Record = ***Current O2 Null Monitor Hourly Record***

else
If (***O2 Wet Monitor Hourly Count*** > 0)
Return result D

Results:

Result	Response	Severity
A	You did not report a MHV record for O2C with a MoistureBasis of W for the hour.	Critical Error Level 1
B	You did not report a MHV record for O2C with a MoistureBasis of W (or blank) for the hour.	Critical Error Level 1
C	You reported too many MHV records for O2C with a MoistureBasis of W (or blank) for the hour.	Critical Error Level 1
D	You reported an MHV record for O2C with a MoistureBasis of [basis], but this record is not required to calculate emissions.	Non-Critical Error
E	You did not report a MHV record for O2C with a MoistureBasis of W for the hour.	Critical Error Level 1
F	You did not report a MHV record for O2C with a MoistureBasis of W (or blank) for the hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-42

Check Name: Verify Single SO2R Derived Hourly Data Record

Related Former Checks:

Applicability: CEM Check

Description: This check scans the DerivedHourlyValueData records to ensure that a single record containing SO2R derived values is reported for the current hour

Specifications:

If (***Derived Hourly Checks Needed*** == true)
SO2R Derived Checks Needed = false

SO2R Derived Hourly Count = count of DerivedHourlyValueData records with ParameterCode = "SO2R" where
Current Date = DerivedHourlyValueData.Date and
Current Hour = DerivedHourlyValueData.Hour

If ***Current Hourly Op Record***.OperatingTime > 0
If (***SO2R Derived Hourly Count*** == 0 AND ***F23 Default Max Value*** is not null)
Return result A
Else if (***SO2R Derived Hourly Count*** > 0 AND ***SO2 F23 Method Active For Hour*** == false)
Return result B
Else if (***SO2R Derived Hourly Count*** > 1)
Return result C
Else if (***SO2R Derived Hourly Count*** == 1)

Current SO2R Derived Hourly Record = DerivedHourlyValueData rec matching with param SO2R where
Current Date = DerivedHourlyValueData.Date and
Current Hour = DerivedHourlyValueData.Hour

SO2R Derived Checks Needed = true

else
If ***SO2R Derived Hourly Count*** > 0
Return result D

Results:

Result	Response	Severity
A	You did not report a DHV record for SO2R for the hour.	Critical Error Level 1
B	You reported a DHV record for SO2R, but this record is not required to calculate emissions.	Non-Critical Error
C	You reported more than one DHV record for SO2R for the hour.	Critical Error Level 1
D	You reported a DHV record for SO2R, but this is not appropriate for a non-operating hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-43

Check Name: Validate Single Stack Flow Record

Related Former Checks:

Applicability: CEM Check

Description: Counts records in MonitorHourlyValueData for the current date and hour with parameter "FLOW". Based on whether or not OperatingTime is non-zero for this hour, reports any appropriate errors and sets Current Flow Monitor Hourly Record

Specifications:

Current Stack Flow Hourly Record= null
Apportionment Stack Flow Array for this Location = null

If (***Flow MHV Optionally Allowed*** == true) AND (***Flow Monitor Hourly Count*** > 0)
Flow Monitor Hourly Checks Needed = true

If ***Flow Monitor Hourly Checks Needed*** == true)
If (***Flow Monitor Hourly Count*** == 0)
Flow Monitor Hourly Checks Needed = false
return result A
Else if (***Flow Monitor Hourly Count*** >1)
return result B
Else
Current Stack Flow Hourly Record = MonitorHourlyValueData record with parameter FLOW where
Current Date = MonitorHourlyValueData.Date and
Current Hour = MonitorHourlyValueData.Hour

Apportionment Stack Flow Array for this Location = ***CurrentStackFlowHourlyRecord***.UnadjustedHourlyValue

else
If ***Flow Monitor Hourly Count*** > 0
return result C

Results:

Result	Response	Severity
A	You did not report an MHV record for FLOW for the hour.	Critical Error Level 1
B	You reported more than one MHV record for FLOW for the hour.	Critical Error Level 1
C	You reported a MHV record for FLOW, but this record is not appropriate for the hour.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-44

Check Name: Check Reporting of Load Range and Common Stack Load Range

Related Former Checks:

Applicability: General Check

Description: Ensures that Load Range and Common Stack Load Range are reported when required and are otherwise not reported.

Specifications:

Set ***CheckLoadRangeValue*** to false.
Set ***CheckCsLoadRangeValue*** to false.

If ***DerivedHourlyChecksNeeded*** == true, AND ***UnitIsLoadBased*** is true,

If ***CurrentHourlyOpRecord***.OpTime > 0, ***CurrentHourlyOpRecord***.HourLoad >= 0, ***LmeAnnual*** is false, AND ***LmeOs*** is false,

If ***FlowMonitorHourlyChecksNeeded*** is true, OR ***NoxConcNeededForNoxMass*** is true, OR ***NoxRateDerivedChecksNeeded*** is true, OR ***So2HpffExists*** is true, OR ***Co2HpffExists*** is true, OR ***HiHpffExists*** is true,

If ***CurrentHourlyOpRecord***.LoadRange is null, AND ***CurrentHourlyOpRecord***.CommonStackLoadRange is null,
return result A.

Else if ***CurrentEntityType*** is equal to "CS",

If ***CurrentHourlyOpRecord***.LoadRange is NOT null,
Set ***CheckLoadRangeValue*** to true.

If ***CurrentHourlyOpRecord***.CommonStackLoadRange is NOT null,

If ***FlowMonitorHourlyCount*** is equal to 0,
return result C.
Else
Set ***CheckCsLoadRangeValue*** to true.

Else if ***CurrentEntityType*** is equal to "CP",

If ***CurrentHourlyOpRecord***.LoadRange is NOT null,
Set ***CheckLoadRangeValue*** to true.

If ***CurrentHourlyOpRecord***.CommonStackLoadRange is NOT null,

If (***HourlyFuelFlowCountOil*** + ***HourlyFuelFlowCountForGas***) is equal to 0,
return result D.
Else
Set ***CheckCsLoadRangeValue*** to true.

Else

If ***CurrentHourlyOpRecord***.LoadRange is NOT null,
Set ***CheckLoadRangeValue*** to true.

If ***CurrentHourlyOpRecord***.CommonStackLoadRange is NOT null,
return result E.

Else

If ***CurrentHourlyOpRecord***.LoadRange is NOT null, OR ***CurrentHourlyOpRecord***.CommonStackLoadRange is NOT null,

return result F.

Else

If ***CurrentHourlyOpRecord***.LoadRange is NOT null, OR ***CurrentHourlyOpRecord***.CommonStackLoadRange is NOT null,

return result B.

Results:

Result	Response	Severity
A	You did not report a Load Range (or Common Stack Load Range) for a monitored operating load-based unit (or associated stack or pipe) that is not in an LME configuration.	Critical Error Level 1
B	You reported a Load Range (or Common Stack Load Range) for a unit (or associated stack or pipe) that was either not operating, not load-based, or is in an LME configuration.	Critical Error Level 1
C	You reported a Common Stack Load Range for a common stack that did not report stack flow.	Critical Error Level 1
D	You reported a Common Stack Load Range for a common pipe that did not report oil or gas fuel flow.	Critical Error Level 1
E	You reported a Common Stack Load Range for a location that is not a common stack or common pipe.	Critical Error Level 1
F	You reported a Load Range (or Common Stack Load Range) for a unit (or associated stack or pipe) that was not monitoring parameters used for substitute data. However, this is not appropriate.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-45

Check Name: Check Reported Load Range Value

Related Former Checks:

Applicability: General Check

Description: Checks the accuracy of the report Load Range if both a Load Range and Load are reported.

Specifications:

Set ***CalculatedLoadRange*** to null.

When ***CheckLoadRangeValue*** is true,

If ***CurrentHourlyOpRecord***.LoadRange is equal to 0,
return result A.

Else if ***CurrentHourlyOpRecord***.HourLoad is NOT null, AND ***CurrentMaximumLoadValue*** is NOT null and > 0,

Set ***CalculatedLoadRange*** to ((10 * ***CurrentHourlyOpRecord***.HourLoad / ***CurrentMaximumLoadValue***) + 1) round down to an integer.

If ***CurrentHourlyOpRecord***.HourLoad is equal to 0,

If ***CurrentHourlyOpRecord***.LoadRange is NOT equal to 1,
return result B.

Else if ***CurrentHourlyOpRecord***.HourLoad >= ***CurrentMaximumLoadValue***,

If ***CurrentHourlyOpRecord***.LoadRange is NOT equal to 10,
return result C.

Else
Set BinSize to (***CurrentMaximumLoadValue*** / 10)

Set *LowRangeBoundry* to (BinSize * (***CurrentHourlyOpRecord***.LoadRange - 1)).
Set *HighRangeBoundry* to (BinSize * ***CurrentHourlyOpRecord***.LoadRange).

If (***CurrentHourlyOpRecord***.HourLoad < *LowRangeBoundry* - 2, OR ***CurrentHourlyOpRecord***.HourLoad > *HighRangeBoundry* + 2,
return result D.

Results:

Result	Response	Severity
A	You should only report a Load Range of 0 (zero) if the load range number is indeterminable.	Critical Error Level 1
B	You reported an Hour Load of 0 (zero), which requires a Load Range of 1 when reported.	Critical Error Level 1
C	You reported an Hour Load that is equal to or exceeds the maximum load, which requires a Load Range of 10 when reported.	Critical Error Level 1
D	You reported a Load Range of [RptLoadRange], but the value calculated using the hourly load and maximum load range is [CalcLoadRange].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-46

Check Name: Check Reported Common Stack Load Range Value

Related Former Checks:

Applicability: General Check

Description: Checks the accuracy of the report Common Stack Load Range if both a Common Stack Load Range and Load are reported.

Specifications:

When ***CheckCsLoadRangeValue*** is true,

If ***CurrentHourlyOpRecord***.CommonStackLoadRange is equal to 0,
return result A.

Else if ***CurrentHourlyOpRecord***.HourLoad is NOT null, AND ***CurrentMaximumLoadValue*** is NOT null and > 0,,

Set ***CalculatedCsLoadRange*** to ((20 * ***CurrentHourlyOpRecord***.HourLoad / ***CurrentMaximumLoadValue***) + 1) round down to an integer.

If ***CurrentHourlyOpRecord***.HourLoad is equal to 0,

If ***CurrentHourlyOpRecord***.CommonStackLoadRange is NOT equal to 1,
return result B.

Else if ***CurrentHourlyOpRecord***.HourLoad >= ***CurrentMaximumLoadValue***,

If ***CurrentHourlyOpRecord***.CommonStackLoadRange is NOT equal to 20,
return result C.

Else
Set BinSize to (***CurrentMaximumLoadValue*** / 20)

Set *LowRangeBoundry* to (BinSize * (***CurrentHourlyOpRecord***.CommonStackLoadRange - 1)).
Set *HighRangeBoundry* to (BinSize * ***CurrentHourlyOpRecord***.CommonStackLoadRange).

If (***CurrentHourlyOpRecord***.HourLoad < *LowRangeBoundry* - 2, OR ***CurrentHourlyOpRecord***.HourLoad < *HighRangeBoundry* + 2,
return result D.

Results:

Result	Response	Severity
A	You should only report a Common Stack Load Range of 0 (zero) if the load range number is indeterminable.	Critical Error Level 1
B	You reported an Hour Load of 0 (zero), which requires a Common Stack Load Range of 1 when reported.	Critical Error Level 1
C	You reported an Hour Load that is equal to or exceeds the maximum load, which requires a Common Stack Load Range of 20 when reported.	Critical Error Level 1
D	You reported a Common Stack Load Range of [RptLoadRange], but the value calculated using the hourly load and maximum load range is [CalcLoadRange].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-47

Check Name: Updated QA Certification Event Supplemental Data

Related Former Checks:

Applicability: General Check

Description: Updates the QA Certification Event Supplemental Data for event dates and conditional data begin dates that occur in the quarter of the emission report being evaluated.

Specifications:

If ***DerivedHourlyChecksNeeded*** is true AND ***CurrentOperatingTime*** is greater than 0,

For each *SupplementalDataRecord* in the dictionary at ***CurrentMonitorPlanLocationPosition*** in ***QaCertEventSuppDataDictionaryArray***,

If *SupplementalDataRecord*.TimeType is equal to "Date",

Increament *SupplementalDataRecord*.QaCertEventQuarterlyOpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) QaCertEventDatehour is in the same quarter as ***CurrentOperatingDatehour***,
3) The date of QaCertEventDatehour is on or before the date of ***CurrentOperatingDatehour***,

Else if *SupplementalDataRecord*.TimeType is equal to "Hour",

Increament *SupplementalDataRecord*.ConditionalDataBeginQuarterlyOpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) ConditionalDataBeginDatehour is not null,
3) ConditionalDataBeginDatehour is in the same quarter as ***CurrentOperatingDatehour***,
4) ConditionalDataBeginDatehour is on or before ***CurrentOperatingDatehour***,

If the month of ***CurrentOperatingDatehour*** is in May or June,.

If *SupplementalDataRecord*.TimeType is equal to "Date",

Increament *SupplementalDataRecord*.QaCertEventQuarterlyOpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) QaCertEventDatehour is in the same quarter as ***CurrentOperatingDatehour***,
3) The date of QaCertEventDatehour is on or before the date of ***CurrentOperatingDatehour***.

Else if *SupplementalDataRecord*.TimeType is equal to "Hour",

Increament *SupplementalDataRecord*.ConditionalDataBeginMayAndJuneOpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) ConditionalDataBeginDatehour is not null,
3) ConditionalDataBeginDatehour is in the same quarter as ***CurrentOperatingDatehour***,
4) ConditionalDataBeginDatehour is on or before ***CurrentOperatingDatehour***.

Results:

Result	Response	Severity

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-48

Check Name: Update System Related Supplemental Data

Related Former Checks:

Applicability: General Check

Description: Updates the System Operating Supplemental Data for Part 75 monitored and derived hourly data, and MATS monitored hourly data.

Specifications:

If ***DerivedHourlyChecksNeeded*** is true AND ***CurrentOperatingTime*** is greater than 0,

Set *SupplementalDataDictionary* to the dictionary at ***CurrentMonitorPlanLocationPosition*** in ***SystemOperatingSuppDataDictionaryArray***.

For each *HourlyRecord* in list:

1) ***CurrentCo2ConcDerivedHourlyRecord***
2) ***CurrentCo2ConcMonitorHourlyRecord***
3) ***CurrentStackFlowHourlyRecord***
4) ***CurrentHeatInputDerivedHourlyRecord***
5) ***CurrentH2oDerivedHourlyRecord***
6) ***CurrentH2oMonitorHourlyRecord***
7) ***CurrentNoxConcMonitorHourlyRecord***
8) ***CurrentNoxRateDerivedHourlyRecord***
9) ***CurrentO2DryMonitorHourlyRecord***
10) ***CurrentO2WetMonitorHourlyRecord***
11) ***CurrentSo2MonitorHourlyRecord***
12) ***MatsHclcMhvRecord***
13) ***MatsHfcMhvRecord***
14) ***MatsHgcMhvRecord***

If *HourlyRecord* is not null, AND *HourlyRecord*.MonitoringSystemID is not null,

If *SupplementalDataDictionary* contains key *HourlyRecord*.MonitoringSystemID,

Set *SupplementalDataRecord* in *SupplementalDataDictionary* value where key is equal to *HourlyRecord*.MonitoringSystemID.

Else

Create a new *SupplementalDataRecord* with MonitoringSystemID equal to *HourlyRecord*.MonitoringSystemID, and OpDays, OpHours, OsDays and OsHours equal to 0.. Add *SupplementalDataRecord* to *SupplementalDataDictionary* with a key of *HourlyRecord*.MonitoringSystemID.

Increament *SupplementalDataRecord*.QuarterlyOperatingCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.QuarterlyOperatingCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.QuarterlyQualityAssuredCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.

2) *HourlyRecord*.ModcCode is in set { 01, 02, 03, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 47, 53, 54 }.

Increament *SupplementalDataRecord*.QuarterlyQualityAssuredCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) *HourlyRecord*.ModcCode is in set { 01, 02, 03, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 47, 53, 54 }.

Increament *SupplementalDataRecord*.QuarterlyMonitorAvailableCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) *HourlyRecord*.ModcCode is in set { 01, 02, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 53 }.

Increament *SupplementalDataRecord*.QuarterlyMonitorAvailableCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) *HourlyRecord*.ModcCode is in set { 01, 02, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 53 }.

Increament *SupplementalDataRecord*.MayAndJuneOperatingCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.

Increament *SupplementalDataRecord*.MayAndJuneOperatingCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.

Increament *SupplementalDataRecord*.MayAndJuneQualityAssuredCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.
3) *HourlyRecord*.ModcCode is in set { 01, 02, 03, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 47, 53, 54 }.

Increament *SupplementalDataRecord*.MayAndJuneQualityAssuredCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.
3) *HourlyRecord*.ModcCode is in set { 01, 02, 03, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 47, 53, 54 }.

Increament *SupplementalDataRecord*.MayAndJuneMonitorAvailableCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.
3) *HourlyRecord*.ModcCode is in set { 01, 02, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 53 }.

Increament *SupplementalDataRecord*.MayAndJuneMonitorAvailableCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.

3) *HourlyRecord*.ModcCode is in set { 01, 02, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 53 }.

For each *SupplementalDataRecord* in the list in the ***QaCertEventSuppDataDictionaryForSystem*** with *HourlyRecord*.MonitoringSystemID as a key,

Increament *SupplementalDataRecord*.QaCertEventQuarterlySystemOpearting.Days by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) QaCertEventDatehour is in the same quarter as ***CurrentOperatingDatehour***,
3) The date of QaCertEventDatehour is on or before the date of ***CurrentOperatingDatehour***,

Increament *SupplementalDataRecord*.ConditionalDataBeginQuarterlySystemOperating.Hours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) ConditionalDataBeginDatehour is not null,
3) ConditionalDataBeginDatehour is in the same quarter as ***CurrentOperatingDatehour***,
4) ConditionalDataBeginDatehour is on or before ***CurrentOperatingDatehour***,

If the month of ***CurrentOperatingDatehour*** is in May or June,.

Increament *SupplementalDataRecord*.QaCertEventMayAndJuneSystemOperating.Days by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) QaCertEventDatehour is in the same quarter as ***CurrentOperatingDatehour***,
3) The date of QaCertEventDatehour is on or before the date of ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.ConditionalDataBeginMayAndJuneSystemOperating.Hours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) ConditionalDataBeginDatehour is not null,
3) ConditionalDataBeginDatehour is in the same quarter as ***CurrentOperatingDatehour***,
4) ConditionalDataBeginDatehour is on or before ***CurrentOperatingDatehour***.

If *HourlyRecord*.MODCCode in set { 01, 02, 03, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 47, 53, 54 },

For each *SupplementalDataRecord* in the list in the ***QaCertEventSuppDataDictionaryForSystem*** with *HourlyRecord*.MonitoringSystemID as a key,

Increament *SupplementalDataRecord*.QaCertEventQuarterlySystemQualityAssuredDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) QaCertEventDatehour is in the same quarter as ***CurrentOperatingDatehour***,
3) The date of QaCertEventDatehour is on or before the date of ***CurrentOperatingDatehour***,

Increament

SupplementalDataRecord.ConditionalDataBeginQuarterlySystemQualityAssuredHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) ConditionalDataBeginDatehour is not null,
3) ConditionalDataBeginDatehour is in the same quarter as ***CurrentOperatingDatehour***,
4) ConditionalDataBeginDatehour is on or before ***CurrentOperatingDatehour***,

If the month of ***CurrentOperatingDatehour*** is in May or June,.

Increament *SupplementalDataRecord*.QaCertEventMayAndJuneSystemQualityAssuredDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) QaCertEventDatehour is in the same quarter as ***CurrentOperatingDatehour***,
3) The date of QaCertEventDatehour is on or before the date of ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.ConditionalDataBeginMayAndJuneSystemQualityAssuredHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) ConditionalDataBeginDatehour is not null,
3) ConditionalDataBeginDatehour is in the same quarter as ***CurrentOperatingDatehour***,
4) ConditionalDataBeginDatehour is on or before ***CurrentOperatingDatehour***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-49

Check Name: Update Component Related Supplemental Data

Related Former Checks:

Applicability: General Check

Description: Updates the Component Operating Supplemental Data for Part 75 and MATS monitored hourly data.

Specifications:

If ***DerivedHourlyChecksNeeded*** is true AND ***CurrentOperatingTime*** is greater than 0,

Set *SupplementalDataDictionary* to the dictionary at ***CurrentMonitorPlanLocationPosition*** in ***ComponentOperatingSuppDataDictionaryArray***.

For each *HourlyRecord* in list:

1) ***CurrentCo2ConcMonitorHourlyRecord***
2) ***CurrentStackFlowHourlyRecord***
3) ***CurrentH2oMonitorHourlyRecord***
4) ***CurrentNoxConcMonitorHourlyRecord***
5) ***CurrentO2DryMonitorHourlyRecord***
6) ***CurrentO2WetMonitorHourlyRecord***
7) ***CurrentSo2MonitorHourlyRecord***
8) ***MatsHclcMhvRecord***
9) ***MatsHfcMhvRecord***
10) ***MatsHgcMhvRecord***

If *HourlyRecord* is not null, AND *HourlyRecord*.ComponentID is not null,

If *SupplementalDataDictionary* contains key *HourlyRecord*.ComponentID,

Set *SupplementalDataRecord* in *SupplementalDataDictionary* value where key is equal to *HourlyRecord*.ComponentID.

Else

Create a new *SupplementalDataRecord* with ComponentID equal to *HourlyRecord*.ComponentID, and OpDays, OpHours, OsDays and OsHours equal to 0..
Add *SupplementalDataRecord* to *SupplementalDataDictionary* with a key of *HourlyRecord*.ComponentID.

Increament *SupplementalDataRecord*.QuarterlyOperatingCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.QuarterlyOperatingCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.QuarterlyQualityAssuredCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) *HourlyRecord*.ModcCode is in set { 01, 02, 03, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 47, 53, 54 }.

Increament *SupplementalDataRecord*.QuarterlyQualityAssuredCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) *HourlyRecord*.ModcCode is in set { 01, 02, 03, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 47, 53, 54 }.

Increament *SupplementalDataRecord*.QuarterlyMonitorAvailableCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) *HourlyRecord*.ModcCode is in set { 01, 02, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 53 }.

Increament *SupplementalDataRecord*.QuarterlyMonitorAvailableCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) *HourlyRecord*.ModcCode is in set { 01, 02, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 53 }.

Increament *SupplementalDataRecord*.MayAndJuneOperatingCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.

Increament *SupplementalDataRecord*.MayAndJuneOperatingCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.

Increament *SupplementalDataRecord*.MayAndJuneQualityAssuredCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.
3) *HourlyRecord*.ModcCode is in set { 01, 02, 03, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 47, 53, 54 }.

Increament *SupplementalDataRecord*.MayAndJuneQualityAssuredCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.
3) *HourlyRecord*.ModcCode is in set { 01, 02, 03, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 47, 53, 54 }.

Increament *SupplementalDataRecord*.MayAndJuneMonitorAvailableCounts.OpDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.
3) *HourlyRecord*.ModcCode is in set { 01, 02, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 53 }.

Increament *SupplementalDataRecord*.MayAndJuneMonitorAvailableCounts..OpHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) The month of ***CurrentOperatingDatehour*** is in May, June, July, August or September.
3) *HourlyRecord*.ModcCode is in set { 01, 02, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 53 }.

For each *SupplementalDataRecord* in the list in the ***QaCertEventSuppDataDictionaryForComponent***

with *HourlyRecord.*ComponentID as a key,

Increament *SupplementalDataRecord*.QaCertEventQuarterlySystemOpearting.Days by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) QaCertEventDatehour is in the same quarter as ***CurrentOperatingDatehour***,
3) The date of QaCertEventDatehour is on or before the date of ***CurrentOperatingDatehour***,

Increament *SupplementalDataRecord*.ConditionalDataBeginQuarterlySystemOpearting.Hours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) ConditionalDataBeginDatehour is not null,
3) ConditionalDataBeginDatehour is in the same quarter as ***CurrentOperatingDatehour***,
4) ConditionalDataBeginDatehour is on or before ***CurrentOperatingDatehour***,

If the month of ***CurrentOperatingDatehour*** is in May or June,.

Increament *SupplementalDataRecord*.QaCertEventMayAndJuneSystemOpearting.Days by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) QaCertEventDatehour is in the same quarter as ***CurrentOperatingDatehour***,
3) The date of QaCertEventDatehour is on or before the date of ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.ConditionalDataBeginMayAndJuneSystemOpearting.Hours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) ConditionalDataBeginDatehour is not null,
3) ConditionalDataBeginDatehour is in the same quarter as ***CurrentOperatingDatehour***,
4) ConditionalDataBeginDatehour is on or before ***CurrentOperatingDatehour***.

If *HourlyRecord.*MODCCode in set { 01, 02, 03, 04, 14, 16, 17, 19, 20, 21, 22, 32, 33, 41, 42, 43, 44, 47, 53, 54 },

For each *SupplementalDataRecord* in the list in the ***QaCertEventSuppDataDictionaryForComponent*** with *HourlyRecord.*ComponentID as a key,

Increament *SupplementalDataRecord*.QaCertEventQuarterlySystemQualityAssuredDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) QaCertEventDatehour is in the same quarter as ***CurrentOperatingDatehour***,
3) The date of QaCertEventDatehour is on or before the date of ***CurrentOperatingDatehour***,

Increament *SupplementalDataRecord*.ConditionalDataBeginQuarterlySystemQualityAssuredHours by 1

when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) ConditionalDataBeginDatehour is not null,
3) ConditionalDataBeginDatehour is in the same quarter as ***CurrentOperatingDatehour***,
4) ConditionalDataBeginDatehour is on or before ***CurrentOperatingDatehour***,

If the month of ***CurrentOperatingDatehour*** is in May or June,.

Increament *SupplementalDataRecord*.QaCertEventMayAndJuneSystemQualityAssuredDays by 1 when:

1) An increament has not already occurred for the date of ***CurrentOperatingDatehour***.
2) QaCertEventDatehour is in the same quarter as ***CurrentOperatingDatehour***,
3) The date of QaCertEventDatehour is on or before the date of ***CurrentOperatingDatehour***.

Increament *SupplementalDataRecord*.ConditionalDataBeginMayAndJuneSystemQualityAssuredHours by 1 when:

1) An increament has not already occurred for ***CurrentOperatingDatehour***.
2) ConditionalDataBeginDatehour is not null,
3) ConditionalDataBeginDatehour is in the same quarter as ***CurrentOperatingDatehour***,
4) ConditionalDataBeginDatehour is on or before ***CurrentOperatingDatehour***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-50

Check Name: Update Last Quality Assured Supplemental Data

Related Former Checks:

Applicability: General Check

Description: Updates the Last Quality Assured Supplemental Data for Part 75 monitord and derived hourly data.

Specifications:

If ***DerivedHourlyChecksNeeded*** is true AND ***CurrentOperatingTime*** is greater than 0,

Set *SupplementalDataDictionary* to the dictionary at ***CurrentMonitorPlanLocationPosition*** in ***LastQualityAssuredSuppDataDictionaryArray***.

For each *HourlyRecord* in list:

1) ***CurrentCo2ConcDerivedHourlyRecord***
2) ***CurrentCo2ConcMonitorHourlyRecord***
3) ***CurrentStackFlowHourlyRecord***
4) ***CurrentH2oDerivedHourlyRecord***
5) ***CurrentH2oMonitorHourlyRecord***
6) ***CurrentNoxConcMonitorHourlyRecord***
7) ***CurrentNoxRateDerivedHourlyRecord***
8) ***CurrentO2DryMonitorHourlyRecord***
9) ***CurrentO2WetMonitorHourlyRecord***
10) ***CurrentSo2MonitorHourlyRecord***

If *HourlyRecord* is ***CurrentCo2ConcDerivedHourlyRecord*** or ***CurrentH2oDerivedHourlyRecord*** or ***CurrentNoxRateDerivedHourlyRecord***,

Set *HourlyTypeCode* to "DERIVED".
Set *MoistureBasis* to null.
Set *ComponentKey* to null.

Otherwise

Set *HourlyTypeCode* to "MONITOR".
Set *MoistureBasis* to *HourlyRecord*.MoistureBasis.
Set *ComponentKey* to *HourlyRecord*.ComponentKey .

If *HourlyRecord* is not null, AND *HourlyRecord*.ModcCode in *QualityAssuredModcSet* ,

For three passes with the following condition and settings for each pass:

Pass 1) *MonitorSystemTarget* equal to null and *ComponentTarget* equal to null.
Pass 2) Run when *HourlyRecord*.MonitorSystemKey is not null with *MonitorSystemTarget* equal to *HourlyRecord*.MonitorSystemKey and *ComponentTarget* equal to null.
Pass 3) Run when *ComponentKey* is not null with *MonitorSystemTarget* equal to null and *ComponentTarget* equal to *ComponentKey*.

Set *QualityAssuredKey* to the concatenation of:

1) ***CurrentOperatingDatehour***
2) *HourlyRecord*.MonitorLocationKey
3) *HourlyRecord*.ParameterCode,

4) *MoistureBasis*,
5) *HourlyTypeCode*,
6) *MonitorSystemTarget*.
7) *ComponentTarget*.

If *SupplementalDataDictionary* contains key *QualityAssuredKey*,

Set *SupplementalDataRecord* to *SupplementalDataDictionary* value where key is equal to *QualityAssuredKey*.

Else

Create a new *SupplementalDataRecord* with:

1) MonitorLocationKey equal to *HourlyRecord*.MonitorLocationKey .
2) ReportingPeriodKey equal to *HourlyRecord*.ReportingPeriodKey .
3) ParameterCode equal to *HourlyRecord*.ParameterCode.
4) MoistureBasis equal to *MoistureBasis*.
5) HourlyTypeCode equal to *HourlyTypeCode*.
6) MonitorSystemKey equal to *MonitorSystemTarget*.
7) ComponentKey to *ComponentTarget*.
8) UnadjustedHourlyValue equal to null.
9) AdjustedHourlyValue equal to null.

Add *SupplementalDataRecord* to *SupplementalDataDictionary* with a key of *QualityAssuredKey*.

Set *SupplementalDataRecord*.UnadjusteHourlyValue to *HourlyRecord*.UnadjusteHourlyValue.
Set *SupplementalDataRecord*.AdjusteHourlyValue to *HourlyRecord*.AdjusteHourlyValue.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-51

Check Name: Set Primary Bypass Information

Related Former Checks:

Applicability: General Check

Description: Set check parameters used in the processing of primary bypass system (stacks) and associated primary systems (stacks) when primary bypass systems are involved for an emission report.

Specifications:

Set ***PrimaryBypassActiveForHour*** to false.
Set ***PrimaryBypassActivePrimarySystemId*** to null.
Set ***PrimaryBypassActiveBypassSystemId*** to null.

If ***PrimaryBypassActiveInQuarter*** is true,

Locate *BypassSystemRecord* in ***MonitorSystemRecordsByHourLocation*** where:

1) SystemTypeCode equals "NOX".
2) SystemDesignationCode equals "PB".

If found,

Set ***PrimaryBypassActiveForHour*** to true.
Set ***PrimaryBypassActiveBypassSystemId*** to *BypassSystemRecord*.MonitorSystemId.

Locate *PrimarySystemRecord* in ***MonitorSystemRecordsByHourLocation*** where:

1) SystemTypeCode equals "NOX".
2) SystemDesignationCode equals "P".

If found,

Set ***PrimaryBypassActivePrimarySystemId*** to *PrimarySystemRecord*.MonitorSystemId.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: HOUROP-52

Check Name: Update Daily Calibration Operating Information

Related Former Checks:

Applicability: General Check

Description: Updates the operating hour count, last covered non-op hour, and first op hour after last covered non--op hour information for daily calibration tests being tracked for status checking.

Specifications:

If ***DerivedHourlyChecksNeeded*** is true,

Initialize *BypassSystemActiveSystemOpTimeDictionary*.

If (***PrimaryBypassActiveForHour*** is true)

If (***CurrentNoxRateDerivedHourlyRecord*** is NOT null) and (***CurrentNoxRateDerivedHourlyRecord*** MonitoringSystemId is NOT null, SystemTypeCode equals "NOX" and SystemDesignationCode equals "P" or "PB")

Add ***CurrentNoxRateDerivedHourlyRecord*** MonitoringSystemId to *BypassSystemActiveSystemOpTimeDictionary* with a value of ***CurrentHourlyOpRecord***.OperatingTime.

For each *UnusedMhvRecord* in ***NoxrPrimaryOrPrimaryBypassMhvRecords***,

If (*UnusedMhvRecord*.NotReportedNoxMonitoringSystemID is NOT in *BypassSystemActiveSystemOpTimeDictionary*)

Add *UnusedMhvRecord*.NotReportedNoxMonitoringSystemID to *BypassSystemActiveSystemOpTimeDictionary* with a value of ***CurrentHourlyOpRecord***.OperatingTime.

If (***PrimaryBypassActiveBypassSystemId*** is NOT in *BypassSystemActiveSystemOpTimeDictionary*)

Add ***PrimaryBypassActiveBypassSystemId*** to *BypassSystemActiveSystemOpTimeDictionary* with a value of 0.

If (***PrimaryBypassActivePrimarySystemId*** is NOT null) and (***PrimaryBypassActivePrimarySystemId*** is NOT in *BypassSystemActiveSystemOpTimeDictionary*)

Add ***PrimaryBypassActivePrimarySystemId*** to *BypassSystemActiveSystemOpTimeDictionary* with a value of 0.

For each *TestLocationObject* in ***MostRecentDailyCalibratonTestObject*** for the location,

If (***CurrentHourlyOpRecord***.OperatingTime is greater than 0)

Increament *TestLocationObject*.OperatingHourCount by 1.

If (*TestLocationObject*.LastCoveredNonOpHour is NOT null) AND (*TestLocationObject*.FirstOpHourAfterLastCoveredNonOpHour is null)

Set *TestLocationObject*.FirstOpHourAfterLastCoveredNonOpHour to ***CurrentHourlyOpRecord***.Date/Hour.

Else

If (the number of inclusive days from *TestLocationObject*.DailyTestDateHour through ***CurrentHourlyOpRecord***.Date/Hour is inclusively between 1 and 26)

Set *TestLocationObject*.LastCoveredNonOpHour to ***CurrentHourlyOpRecord***.Date/Hour.
Set *TestLocationObject*.FirstOpHourAfterLastCoveredNonOpHour to null.

For each *SystemOpTimeEntry* in *BypassSystemActiveSystemOpTimeDictionary* for the location,

Set *TestSystemObject* to the entry in *TestLocationObject*.SystemSupplementalValuesDictionary where key equals *SystemOpTimeEntry*.MonitorSystemId.

If (*SystemOpTimeEntry*.OpTime is greater than 0)

Increament *TestSystemObject* .OperatingHourCount by 1.

If (*TestSystemObject* .LastCoveredNonOpHour is NOT null) AND (*TestSystemObject* .FirstOpHourAfterLastCoveredNonOpHour is null)

Set *TestSystemObject* .FirstOpHourAfterLastCoveredNonOpHour to ***CurrentHourlyOpRecord***.Date/Hour.

Else

If (the number of inclusive days from *TestSystemObject* .DailyTestDateHour through ***CurrentHourlyOpRecord***.Date/Hour is inclusively between 1 and 26)

Set *TestSystemObject* .LastCoveredNonOpHour to ***CurrentHourlyOpRecord***.Date/Hour.
Set *TestSystemObject* .FirstOpHourAfterLastCoveredNonOpHour to null.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-53

Check Name: Update Daily Interference Operating Information

Related Former Checks:

Applicability: General Check

Description: Updates the operating hour count, last covered non-op hour, and first op hour after last covered non--op hour information for daily interference tests tracked for status checking.

Specifications:

If ***DerivedHourlyChecksNeeded*** is true,

For each *TestObject* in ***LatestDailyInterferenceCheckObject*** for ***CurrentMonitorLocationId***,

If (***CurrentOperatingTime*** is greater than 0)

Increament *TestObject*.OperatingHourCount by 1.

If (*TestObject*.LastCoveredNonOpHour is NOT null) AND (*TestObject*.FirstOpHourAfterLastCoveredNonOpHour is null)

Set *TestObject*.FirstOpHourAfterLastCoveredNonOpHour to ***CurrentOperatingDatehour***.

Else

If (the number of inclusive days from *TestObject*.DailyTestDateHour through ***CurrentOperatingDatehour*** is inclusively between 1 and 26)

Set *TestObject*.LastCoveredNonOpHour to ***CurrentOperatingDatehour***.
Set *TestObject*.FirstOpHourAfterLastCoveredNonOpHour to null.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Unit-Level Evaluation

Check Code: HOUROP-54

Check Name: Check for Linearity Tests During Non-Op Hours

Related Former Checks:

Applicability: CEM Check

Description: Returns a result when a location did not operate but a liniearity exists with a begin or end hour equal to the current operating hour.

Specifications:

Set ***LinearityOfflineList*** to "".

If ***DerivedHourlyChecksNeeded*** is true AND ***CurrentOperatingTime*** equals 0.00,

If ***LinearityExistsLocationArray*** value at ***CurrentMonitorPlanLocationPosition*** is true,

For each *LinearitySuppDataRecord* in ***LinearityTestRecordsByLocationForQaStatus*** where:

1) MonitorLocationId equals ***CurrentMonitorLocationId***.
2) BeginDate/BeginHour OR EndDate/EndHour equals ***CurrentOperatingDateHour***.

Append *LinearitySuppDataRecord*.TestNumber to ***LinearityOfflineList***.

If ***LinearityOfflineList*** does not equal "",

Return result A.

Results:

Result	Response	Severity
A	Linearity and System Integrity tests are not allowed during non-operating hours, but the begin or end hour of the following test(s) occurred during this non-operating hour: [list].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Category:

Leak Status

Check Code: LKSTAT-1

Check Name: Locate Most Recent Prior Leak Check

Related Former Checks:

Applicability: CEM Check

Description: Determines if there is an applicable prior Leak Check.

Specifications:

Set ***PriorLeakRecord*** = null.

Locate the most recent record in ***LeakCheckRecordsByLocationForQAStatus*** for the location where:

a) the ComponentID is equal to the ***QaStatusComponentId***
b) the TestResultCd is not equal to "INVALID" and
c) the EndDate/Hour is on or prior to the ***CurrentMHVRecord***.Date/Hour

if (***LeakCheckRecordsByLocationForQAStatus*** is found)

Set ***PriorLeakRecord*** = the found record in ***LeakCheckRecordsByLocationForQAStatus***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Leak Check Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Leak Status Evaluation

Check Code: LKSTAT-2

Check Name: Locate Most Recent Prior Event

Related Former Checks:

Applicability: CEM Check

Description: Determines if there is an applicable prior event.

Specifications:

Set ***PriorLeakEventRecord*** = null.
Set ***LeakStatusResult*** = null

If (***PriorLeakRecord*** is null)

Locate the latest record in ***QACertificationEventRecords*** where the ComponentID is equal to the ***QaStatusComponentId***, QaCertEventCode is equal to "300" or "305", and the QACertEventDate/Hour is prior to ***CurrentReportingPeriod***.

if (***QACertificationEventRecord*** is found)

Set ***PriorLeakEventRecord*** = the found record in ***QACertificationEventRecord***.

Locate the latest record in ***RataTestRecordsByLocationForQaStatus*** where the SystemID is equal to ***CurrentMhvRecord***.SystemID, TestReasonCode is equal to "INITIAL" or "RECERT", TestResultCode is equal to "PASSED", and EndDateHour is after ***QACertificationEventRecord***.QACertEventDate/Hour and is prior to ***CurrentReportingPeriod***.

if found,
Set ***LeakStatusResult*** = "OOC-Event".

else
Set ***LeakStatusResult*** = "IC".

else
Locate the ***SystemComponent*** record with the earliest BeginDate where the ComponentID is equal to the ***QaStatusComponentId***.

If found, and the BeginDate in the ***SystemComponent*** record is in the current reporting period,
Set ***LeakStatusResult*** = "IC"

else
Set ***LeakStatusResult*** = "OOC-No Prior Test".

else

Locate the latest record in ***QACertificationEventRecords*** where the ComponentID is equal to the ***QaStatusComponentId***, LeakRequired is equal to "Y" and QaCertEventCode is not equal to "300", and the QACertEventDate/Hour is prior to the ***CurrentMHVRecord***.Date/Hour and after the ***PriorLeakRecord***.EndDate/Hour,

if a record is found
Set ***PriorLeakEventRecord*** = the found record in ***QACertificationEventRecords***.

if ***PriorLeakEventRecord***.LastCompletedTestDate is not null
Set *ExpectedLeakCheckQuarter* = the quarter of ***PriorLeakEventRecord***.LastCompletedTestDate.
else
Set *ExpectedLeakCheckQuarter* = the quarter after ***PriorLeakEventRecord***.QACertEventDate.

Set *QuartersAfterCount = 0.*

For each quarter beginning with *ExpectedLeakCheckQuarter* and ending with ***CurrentReportingPeriod***,

if *QuartersAfterCount* is equal to 4,
Set *RequiredLeakCheckQuarter* to quarter.
Exit the loop.

else if ***AnnualReportingRequirement*** equals true, or the quarter being checked is 2 or 3

if ***AnnualReportingRequirement*** equals true, or the quarter being checked is 3
Locate the record in ***OperatingSuppDataRecordsByLocation*** where OpTypeCode is equal to "OPHOURS", FuelCode is null, and reporting period equals the quarter being checked
else
Locate the record in ***OperatingSuppDataRecordsByLocation*** where OpTypeCode is equal to "OSHOURS", FuelCode is null, and reporting period equals the quarter being checked

if ***OperatingSuppDataRecordsByLocation*** record is found

If ***OperatingSuppDataRecordsByLocation***.OpValue >= 168
Set *RequiredLeakCheckQuarter* to quarter.
Exit the loop.

else
Set *RequiredLeakCheckQuarter* to quarter.
Exit the loop.

Increament *QuartersAfterCount by 1.*

if *RequiredLeakCheckQuarter* is prior to ***CurrentReportingPeriod***
Set ***LeakStatusResult*** ="OOC-Event".
else
Set ***LeakStatusResult*** = "IC".

else if ***PriorLeakRecord***.QANeedsEvaluationFlag = "Y"
Set ***LeakStatusResult*** = "Prior Test Not Yet Evaluated".

else if ***PriorLeakRecord***.TestResultCd is null
Set ***LeakStatusResult*** = "OOC-Test Has Critical Errors".

else if ***PriorLeakRecord***.TestResultCd == "FAILED"
Set ***LeakStatusResult*** = "OOC-Test Failed".

else if ***PriorLeakRecord***.TestResultCd == "ABORTED"
Set ***LeakStatusResult*** = "OOC-Test Aborted".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Leak Check Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Leak Status Evaluation

Check Code: LKSTAT-3

Check Name: Determine Expiration Date For Prior Leak Check

Related Former Checks:

Applicability: CEM Check

Description: Determines if Leak Check occurred prior to the standard expiration date.

Specifications:

if (***LeakStatusResult*** is null)

Set ***PriorLeakExpirationDate*** = ***PriorLeakRecord***.TestExpirationDate.

if (***PriorLeakExpirationDate*** is null)

if (***Annual Reporting Requirement*** == false)

if (***PriorLeakRecord*** .TestEndQuarter = "2")

Set ***PriorLeakExpirationDate*** = September 30th following ***PriorLeakRecord*** .EndDate.

else

Set ***PriorLeakExpirationDate*** = June 30th following ***PriorLeakRecord*** .EndDate.

else

if (***PriorLeakRecord*** .GracePeriodInd = 1)

Set ***PriorLeakExpirationDate*** = the end of the quarter of the ***PriorLeakRecord***.EndDate.

else

Set ***PriorLeakExpirationDate*** = the end of the quarter following the quarter of the ***PriorLeakRecord*** .EndDate.

Set ***PriorLeakRecord*** .TestExpirationDate = ***PriorLeakExpirationDate*** .

if (***CurrentMHVRecord***.Date is ON OR BEFORE the ***PriorLeakExpirationDate***)

Set ***LeakStatusResult*** = "IC".

else

Set ***PriorLeakExpirationDate*** = null.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Leak Check Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Leak Status Evaluation

Check Code: LKSTAT-4

Check Name: Determine Extended Expiration Date for Prior Leak Check

Related Former Checks:

Applicability: CEM Check

Description: Determines if Leak Check occurred prior to the extended expiration date.

Specifications:

Set ***LeakMissingOpDataInfo*** = null.

if (***LeakStatusResult*** is null)

Set *MissingOpData* = false

if (***PriorLeakRecord***.TestExpirationDateWithExtension is null)

For each quarter beginning with the quarter of the ***PriorLeakExpirationDate*** and continuing through the quarter prior to the quarter of the ***CurrentMHVRecord***.Date:

if (***EarliestLocationReportDate*** > the last day of the quarter being checked)

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.

else

If (***Annual Reporting Requirement*** == true OR the quarter being checked is == 2 or 3)

If (***Annual Reporting Requirement*** == true OR the quarter being checked == 3)

Locate a record in ***OperatingSuppDataRecordsByLocation*** where the reporting period is equal to the year/quarter being checked and the OpTypeCode = "OPHOURS" and FuelCode is null.

else

Locate a record in ***OperatingSuppDataRecordsByLocation*** where the reporting period is equal to the year/quarter being checked and the OpTypeCode = "OSHOURS" and FuelCode is null.

if (***OperatingSuppDataRecordsByLocation*** is found AND ***OperatingSuppDataRecordsByLocation***.OpValue < 168)

If (***Annual Reporting Requirement*** = true OR the quarter being checked == 2)

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.

else

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 3.

else if (***OperatingSuppDataRecordsByLocation*** is not found)

If (the quarter being checked == 1 or 4)

Locate a ***LocationReportingFrequency*** record for the test location where ReportingFrequencyCode = "OS", the Begin Quarter is on or before the quarter being checked and the EndQuarter is null or is on or after the quarter being

checked.

If (***LocationReportingFrequency*** record is found)

if (***Annual Reporting Requirement*** == true and the quarter being checked == 4 and the year of the EndQuarter is equal to the year of the quarter being checked.

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.

else

Stop looking for extensions.

else

Set *Missing Op Data* to true.
Set ***LeakMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
Stop looking for extensions.

else

Set *Missing Op Data* to true.
Set ***LeakMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
Stop looking for extensions.

else

Stop looking for extensions.

Add *NumberOfExtensionQuarters* to ***PriorLeakExpirationDate***
Set ***PriorLeakRecord***.TestExpirationDateWithExtension = ***PriorLeakExpirationDate***

else
Set ***PriorLeakExpirationDate*** = ***PriorLeakRecord***.TestExpirationDateWithExtension

If (***CurrentMHVRecord***.Date/Hour is ON OR BEFORE the ***PriorLeakExpirationDate***)

Set ***LeakStatusResult*** = "IC-Extension".

else if (*Missing Op Data* == true)

Set ***LeakStatusResult*** = "Missing Op Data".
Set ***PriorLeakRecord***.TestExpirationDateWithExtension = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Leak Check Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Leak Status Evaluation

Check Code: LKSTAT-5

Check Name: Determine Grace Period for Leak Check

Related Former Checks:

Applicability: CEM Check

Description: Determines if Leak Check occurred prior to the end of the grace period.

Specifications:

if (***LeakStatusResult*** is null)

if (***Annual Reporting Requirement*** == false)

Set ***LeakStatusResult*** = "OOC-Expired".

else if (***Rpt Period Op Hours Accumulator Array*** for the location == -1)

Set ***LeakStatusResult*** = "Invalid Op Data".

else

Set *GraceOpHours* = ***RptPeriodOpHoursAccumulatorArray*** for the location.

if (*GraceOpHours* > 168)

Set ***LeakStatusResult*** = "OOC-Expired".

else

If the quarter after the LATER of the ***PriorLeakExpirationDate*** and the ***EarliestLocationReportDate*** is the quarter of the ***CurrentMHVRecord***.Date/Hour ,

Set ***LeakStatusResult*** = "IC-Grace".

else

For each quarter beginning with the quarter after the LATER of the ***PriorLeakExpirationDate*** and the ***EarliestLocationReportDate*** ,and continuing through the quarter prior to the ***CurrentMHVRecord***.Date/Hour ,

if (***EarliestLocationReportDate*** <= the last day of the quarter being checked)

Locate a record in ***OperatingSuppDataRecordsByLocation*** where the reporting period is equal to the year/quarter being checked and the OpTypeCode = "OPHOURS" and FuelCode is null.

if (***OperatingSuppDataRecordsByLocation*** is found)

Add OpValue to *GraceOpHours.*

if (*GraceOpHours* > 168)

Set ***LeakStatusResult*** = "OOC-Expired".
exit for.

else

Set ***LeakStatusResult*** = "Missing Op Data".

Set ***LeakMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
exit for.

if (***LeakStatusResult*** is null)
Set ***LeakStatusResult*** = "IC-Grace".

if ***LeakStatusResult*** does not begin with "IC"
return ***LeakStatusResult***

Results:

Result	Response	Severity
Invalid Op Data	The [testtype] status for [key] could not be determined, because the OperatingTime in at least one Hourly Operating Data records was missing or invalid.	Critical Error Level 1
Missing Op Data	The Leak Check status for [COMPID] could not be determined, because the Op Supp Data record for OPHOURS or OSHOURS is missing for [MISSINGOPDATAINFO] (and possibly other previous reporting periods). If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
OOC-Event	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for [compid], but you did not perform a subsequent leak check.	Critical Error Level 1
OOC-Expired	The prior leak check for [compid] completed on [date] has expired.	Critical Error Level 1
OOC-No Prior Test	You did not report a prior [testtype] for [key].	Critical Error Level 1
OOC-Test Aborted	The prior leak check for [compid] completed on [date] was aborted.	Critical Error Level 1
OOC-Test Failed	The prior leak check for [compid] completed on [date] failed.	Critical Error Level 1
OOC-Test Has Critical Errors	The prior leak check for [compid] completed on [date] has critical errors.	Critical Error Level 1
Prior Test Not Yet Evaluated	The [testtype] status for [key] could not be determined, because the applicable prior [testtype] with TestNumber [testnum] has not yet been evaluated.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Leak Check Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report ---------- Flow Averaging Leak Status Evaluation

Check Category:

Linearity Status

Check Code: LINSTAT-1

Check Name: Check Analyzer Range Exemption For Linearity Status

Related Former Checks:

Applicability: CEM Check

Description: This check determines if the Current Analyzer Range used is exempt for Linearity Status purposes.

Specifications:

Set ***CurrentLinearityStatus*** = null

if (***CurrentMHVParameter*** = "SO2C" or ***CurrentMHVParameter*** = "NOXC")

Locate the record in ***MonitorSpanRecordsByHourLocation*** for the hour and location where the ComponentTypeCode is equal to the ***QaStatusComponentTypeCode*** and the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed***.

if (***MonitorSpanRecordsByHourLocation*** is not found OR more than one ***MonitorSpanRecordsByHourLocation*** is found OR if the ***MonitorSpanRecordsByHourLocation***.SpanValue is null or <= 0)

Set CurrentLinearityStatus = "Invalid Monitor Span".

else if (***MonitorSpanRecordsByHourLocation*** is found and ***MonitorSpanRecordsByHourLocation***.SpanValue <= 30)

Set ***CurrentLinearityStatus*** = "IC-Exempt".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2 Linearity Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- Hg Linearity Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- NOX Linearity Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB CO2C, NOXC or O2C Linearity St
5	Process/Category:	Emissions Data Evaluation Report --------- O2 Dry Linearity Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- O2 Wet Linearity Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- SO2 Linearity Status Evaluation

Check Code: LINSTAT-2

Check Name: Locate Most Recent Prior Linearity Test

Related Former Checks:

Applicability: CEM Check

Description: Determines if there is an applicable prior Linearity test.

Specifications:

Set ***PriorLinearityRecord*** = null.
Set ***InvalidLinearityRecord*** = null.

if (***CurrentLinearityStatus*** is null)

Locate the most recent record in ***LinearityTestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***ApplicableComponentID*** and the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed*** and the CalculatedTestResult is not equal to "INVALID" and the EndDate/Hour is either:

a) prior to the ***CurrentDateHour*** OR
b) equal to the ***CurrentDateHour*** and the EndMinute is less than 45 and the CalculatedTestResult is equal to "PASSED" or "PASSAPS".

if (***LinearityTestRecordsByLocationForQAStatus*** is found)

Set ***PriorLinearityRecord*** = the found record in ***LinearityTestRecordsByLocationForQAStatus***.

Locate the most recent record in ***LinearityTestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***ApplicableComponentID*** and the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed*** and the EndDate/Hour is prior to the ***CurrentDateHour*** and the EndDate/Hour is greater than the ***PriorTestRecord***.EndDate/Hour and the CalculatedTestResult is equal to "INVALID".

if (***LinearityTestRecordsByLocationForQAStatus*** is found)

Set ***InvalidLinearityRecord*** = the found record in ***LinearityTestRecordsByLocationForQAStatus***.

else

Locate the most recent record in ***LinearityTestRecordsByLocationForQAStatus*** for the location where the ComponentID is equal to the ***ApplicableComponentID*** and the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed*** and the CalculatedTestResult is equal to "INVALID" and the EndDate/Hour is prior to the ***CurrentDateHour***.

if (***LinearityTestRecordsByLocationForQAStatus*** is found)

Set ***InvalidLinearityRecord*** = the found record in ***LinearityTestRecordsByLocationForQAStatus***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2 Linearity Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- Hg Linearity Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- NOX Linearity Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB CO2C, NOXC or O2C Linearity St
5	Process/Category:	Emissions Data Evaluation Report --------- O2 Dry Linearity Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- O2 Wet Linearity Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- SO2 Linearity Status Evaluation

Check Code: LINSTAT-3

Check Name: Locate Most Recent Prior Event

Related Former Checks:

Applicability: CEM Check

Description: Determines if there is an applicable prior event.

Specifications:

Set ***PriorLinearityEventRecord*** = null.

if (***CurrentLinearityStatus*** is null)

Locate all records in ***QACertificationEventRecords*** where:

the ComponentID is equal to the ***ApplicableComponentID***

AND LinearityRequired is equal to "Y";

AND the QACertEventDate/Hour is either:

a) prior to the ***CurrentDateHour*** OR
b) equal to both the ***CurrentDateHour*** and the ConditionalBeginDate/Hour;

AND either

a) ***Prior Test Record*** is null OR
b) QACertEventDate/Hour is after the ***PriorTestRecord***.EndDate/Hour OR
c) QACertEventDate/Hour is equal to the ***PriorTestRecord***.EndDate/Hour AND (TestCompletionDate is null or the TestCompletionDate/Hour is after the ***PriorTestRecord***.EndDate/Hour)

AND either

a) ***DualRangeStatus*** = false OR
b) ***HighRangeComponentID*** <> ***LowRangeComponentID*** OR
c) QACertEventCode <> 27 or 30 or 172 and ***CurrentAnalyzerRangeUsed*** = "H" OR
d) QACertEventCode <> 35 or 171 and ***CurrentAnalyzerRangeUsed*** = "L"

AND either

a) ***Annual Reporting Requirement*** is equal to true OR
b) QACertEventDate/Hour is on or after April 1 of the year of the ***CurrentDateHour***

if (***QACertificationEventRecords*** is found)

Sort ***QACertificationEventRecords*** by QACertEventDate/Hour descending.

For each record in ***QACertificationEventRecords***

Set ***PriorLinearityEventRecord*** = the found record in ***QACertificationEventRecords***.

if (***PriorLinearityEventRecord***.QACertEventCode = 170 and ***DualRangeStatus*** = true)

Locate the record in ***MonitorSpanRecordsByLocation*** where the ComponentTypeCode is equal to ***QaStatusComponentTypeCode*** and the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed*** and the BeginDate/Hour is equal to the ***PriorLinearityEventRecord.***QACertEventDate/Hour.

if (***MonitorSpanRecordsByLocation*** is found)

exit for loop.

else

set ***PriorLinearityEventRecord*** = null.

else

exit for loop.

If (***PriorLinearityEventRecord*** is null)

If (***PriorLinearityRecord*** is null)

Set ***CurrentLinearityStatus*** = "OOC-No Prior Test or Event".

else if (***InvalidLinearityRecord*** is not null AND***PriorLinearityEventRecord***.QACertEventDate/Hour is after ***InvalidTestRecord***.EndDate/Hour)

Locate the earliest record in ***LinearityTestRecordsByLocationForQAStatus*** where the ComponentID is equal to the ***ApplicableComponentID,*** the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed***, the CalculatedTestResult is equal to "INVALID", and the EndDate/Hour is after the ***PriorLinearityEventRecord***.QACertEventDate/Hour and prior to the ***CurrentDateHour***.

if (***LinearityTestRecordsByLocationForQAStatus*** is found)

Set ***InvalidLinearityRecord*** = the found record in ***LinearityTestRecordsByLocationForQAStatus***.

else

Set ***InvalidLinearityRecord*** = null.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2 Linearity Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- Hg Linearity Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- NOX Linearity Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB CO2C, NOXC or O2C Linearity St
5	Process/Category:	Emissions Data Evaluation Report --------- O2 Dry Linearity Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- O2 Wet Linearity Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- SO2 Linearity Status Evaluation

Check Code: LINSTAT-4

Check Name: Determine Expiration Dates For Most Recent Prior Linearity Test

Related Former Checks:

Applicability: CEM Check

Description: Determines the expiration dates for the Applicable Prior Linearity test. This includes the Test Expiration Date both with and without any extensions

Specifications:

Set ***LinearityMissingOpDataInfo*** = null.

if (***CurrentLinearityStatus*** is blank and ***PriorLinearityRecord*** is not null and ***PriorLinearityEventRecord*** is null)

Set ***CheckForIgnoredLinearity*** = true.
Set *PriorTestExpirationDate* = null
Set *PriorTestExpirationDateWithExtension* = null
Set *MissingOpData* = false

if (***PriorLinearityRecord***.QANeedsEvaluationFlag = "Y")

Set ***CurrentLinearityStatus*** = "Prior Test Not Yet Evaluated".

else if (***PriorLinearityRecord***.TestResultCode = null or ***PriorLinearityRecord***.TestResultCode = "FAILED" or ***PriorLinearityRecord***.TestResultCode = "ABORTED")

Locate the most recent record in ***QACertificationEventRecords*** where:

the ComponentID is equal to the ***ApplicableComponentID***

AND LinearityRequired is equal to "Y";

AND the ConditionalBeginDate/Hour is:

a) on or prior to the ***CurrentDateHour*** AND
b) on or after the ***PriorTestRecord***.EndDate/Hour;

AND either

a) ***DualRangeStatus*** = false OR
b) ***HighRangeComponentID*** <> ***LowRangeComponentID*** OR
c) QACertEventCode <> 27 or 30 or 172 and ***CurrentAnalyzerRangeUsed*** = "H" OR
d) QACertEventCode <> 35 or 171 and ***CurrentAnalyzerRangeUsed*** = "L"

AND either

a) ***Annual Reporting Requirement*** is equal to true OR
b) QACertEventDate/Hour is on or after April 1 of the year of the ***CurrentHourlyRecordforRATAStatus***.Date

if (***QACertificationEventRecords*** is found)

Set ***PriorLinearityEventRecord*** = found record in ***QACertificationEventRecords***.

elseif (***PriorLinearityRecord***.TestResultCode = null)

Set ***CurrentLinearityStatus*** = "OOC-Test Has Critical Errors".

else if (***PriorLinearityRecord***.TestResultCode = "FAILED")

Set ***CurrentLinearityStatus*** = "OOC-Test Failed".

else if (***PriorLinearityRecord***.TestResultCode = "ABORTED")

Set ***CurrentLinearityStatus*** = "OOC-Test Aborted".

else

Set *PriorTestExpirationDate* = ***PriorLinearityRecord***.TestExpirationDate.
Set *PriorTestExpirationDateWithExtension* = ***PriorLinearityRecord***.TestExpirationDateWithExtension.

if (*PriorTestExpirationDate* is null)

if (***Annual Reporting Requirement*** == false)

if (***PriorLinearityRecord***.TestEndQuarter = "2")

Set *PriorTestExpirationDate* = July 30th following ***PriorLinearityRecord***.EndDate.

else
Set *PriorTestExpirationDate* = April 30th following ***PriorLinearityRecord***.EndDate.

else

Set *AlternateTestDate* = null

if (***PriorLinearityRecord***.ComponentTypeCode is equal to "HG")

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the ProgramCode is equal to MATS and the EmissionsRecordingBeginDate is ON OR BEFORE the ***QaStatusComponentBeginDate***.

if found
Set *AlternateTestDate* = EmissionsRecordingBeginDate of the located record

Locate the most recent ***QACertificationEventRecords*** for the ***ApplicableComponentID*** where LinearityRequired is equal to "Y", and the BeginDate/Hour is prior to the ***PriorLinearityRecord***.BeginDate/Hour.

if (***QACertificationEventRecords*** is found AND the LinearityCertEvent is equal to "Y" and the ConditionalDataBeginDate is null and the CompletionTestDate/Hour is after the ***PriorLinearityRecord***.EndDate/Hour)

If *AlternateTestDate* is null

Set *PriorTestExpirationDate* = the end of the quarter following the quarter of the ***QACertificationEventRecords***.CompletionTestDate.

Else
Set *PriorTestExpirationDate* = the end of the quarter following the quarter of the later of ***QACertificationEventRecords***.CompletionTestDate and *AlternateTestDate* .

else if (***PriorLinearityRecord***.GracePeriodInd = 1)

If *AlternateTestDate* is null

Set *PriorTestExpirationDate* = the end of the quarter of the quarter of the ***PriorLinearityRecord***.EndDate.

Else

Set *PriorTestExpirationDate* = the end of the quarter of the quarter of the later of ***PriorLinearityRecord***.EndDate and *AlternateTestDate*.

else

If *AlternateTestDate* is null

Set *PriorTestExpirationDate* = the end of the quarter following the quarter of the ***PriorLinearityRecord***.EndDate.

Else

Set *PriorTestExpirationDate* = the end of the quarter following the quarter of the later of ***PriorLinearityRecord***.EndDate and *AlternateTestDate*.

Set ***PriorLinearityRecord***.TestExpirationDate = *PriorTestExpirationDate*.

if (***CurrentDateHour*** is ON OR BEFORE the *PriorTestExpirationDate*)

Set ***CurrentLinearityStatus*** = "IC".

else if (***Annual Reporting Requirement*** == false)

Set ***CurrentLinearityStatus*** = "OOC-Expired".

else

if (*PriorTestExpirationDateWithExtension* is null)

Set *NumberOfExtensionQuarters* = 0;

For each quarter beginning with the quarter of the *PriorTestExpirationDate* and continuing through the quarter prior to the quarter of the ***CurrentDate***,

if (*NumberOfExtensionQuarters* = 3)

Stop looking for extensions.

else

if (***EarliestLocationReportDate*** > the last day of the quarter being checked)

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.

else

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusComponentTypeCode*** is “CO2”, “NOX” or “O2”)

Locate a record in ***SystemOperatingSuppDataRecordsByLocation*** where:

1) SystemId is equal to

QaStatusPrimaryOrPrimaryBypassSystemId.
2) Year is equal to the year being checked.
3) Quarter is equal to the quarter being checked.
4) OpSuppDataTypeCode = "OP".

If (***SystemOperatingSuppDataRecordsByLocation*** is found)
Set *OperatingHourCount* = ***SystemOperatingSuppDataRecordsByLocation***.Hours.
Else
Set *OperatingHourCount* = null.

Else
Locate a record in ***OperatingSuppDataRecordsByLocation*** where the reporting period is equal to the year/quarter being checked and the OpTypeCode = "OPHOURS".

If (***OperatingSuppDataRecordsByLocation*** is found)
Set *OperatingHourCount* = ***OperatingSuppDataRecordsByLocation***.OpValue.
Else
Set *OperatingHourCount* = null.

if (*OperatingHourCount* is NOT null AND *OperatingHourCount* < 168)

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.

else if (***PriorLinearityRecord***.ComponentTypeCode is NOT equal to "HG")

Locate a record in ***TestExtensionExemptionRecords*** where the ComponentID is equal to the ***ApplicableComponentID*** and the ExtensionExemptionCode is equal to "RANGENU", and the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed*** and the Year/Quarter is equal to the year/quarter to check.

if (***TestExtensionExemptionRecords*** is found)

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.

else

Locate a record in ***TestExtensionExemptionRecords*** where the ComponentID is equal to the ***ApplicableComponentID*** and the ExtensionExemptionCode is equal to "NONQAPB", and the Year/Quarter is equal to the year/quarter to check.

if (***TestExtensionExemptionRecords*** is found)

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.

else if (*OperatingHourCount* is null)

If (the quarter being checked == 1 or 4)

Locate a ***LocationReportingFrequency*** record

for the test location where ReportingFrequencyCode = "OS", the Begin Quarter is on or before the quarter being checked and the EndQuarter is null or is on or before the quarter being checked.

If (***LocationReportingFrequency*** record is found)

if (***Annual Reporting Requirement*** == true and the quarter being checked == 4 and the year of the EndQuarter is equal to the year of the quarter being checked.

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.

else

Stop looking for extensions.

else

Set *Missing Op Data* to true.
Set ***LinearityMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
Stop looking for extensions.

else

Set *Missing Op Data* to true.
Set ***LinearityMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
Stop looking for extensions.

else

Stop looking for extensions.

if (***PriorLinearityRecord***.ComponentTypeCode is NOT equal to "HG")

For each quarter beginning with the quarter after the *End Quarter* and continuing through the quarter prior to the quarter of the ***CurrentDateHour***

Locate a record in ***TestExtensionExemptionRecords*** where the ComponentID is equal to the ***ApplicableComponentID*** and the ExtensionExemptionCode is equal to "NONQAPB", and the Year/Quarter is equal to the year/quarter to check.

if (***TestExtensionExemptionRecords*** is found)

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.

Set *PriorTestExpirationDateWithExtension* = *PriorTestExpirationDate.*
Add *NumberOfExtensionQuarters* to *PriorTestExpirationDateWithExtension.*
Set ***PriorLinearityRecord***.TestExpirationDateWithExtension = *PriorTestExpirationDateWithExtension.*

else

Set ***PriorLinearityRecord***.TestExpirationDateWithExtension = *PriorTestExpirationDate*

If (***CurrentDateHour*** is ON OR BEFORE the *PriorTestExpirationDateWithExtension)*

Set ***CurrentLinearityStatus*** = "IC-Extension".

else if (*Missing Op Data* == true)

Set ***CurrentLinearityStatus*** = "Missing Op Data".
Set ***PriorLinearityRecord***.TestExpirationDateWithExtension = null

else

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusComponentTypeCode*** is “CO2”, “NOX” or “O2”)

Set *CurrentOpHours* = Hours in ***SystemOperatingSuppDataDictionaryArray*** for the current location where SystemId is equal to ***QaStatusPrimaryOrPrimaryBypassSystemId***.

Else

Set *CurrentOpHours* = ***RptPeriodOpHoursAccumulatorArray*** for the location.

if (*CurrentOpHours* == -1)

Set ***CurrentLinearityStatus*** = "Invalid Op Data".

else

Set *GraceOpHours* = *CurrentOpHours* .

if (*GraceOpHours* > 168)

Set ***CurrentLinearityStatus*** = "OOC-Expired".

else

If there are NO quarters beginning with the LATER of the quarter after the *PriorTestExpirationDateWithExtension* and the quarter of the ***EarliestLocationReportDate*** and ending with the quarter prior to the ***CurrentDateHour***,

Set ***CurrentLinearityStatus*** = "IC-Grace".

else

For each quarter beginning with the quarter after the *PriorTestExpirationDateWithExtension* and continuing through the quarter prior to the ***CurrentDateHour***,

if (***EarliestLocationReportDate*** <= the last day of the quarter being

checked)

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusComponentTypeCode*** is “CO2”, “NOX” or “O2”)

Locate a record in ***SystemOperatingSuppDataRecordsByLocation*** where:

1) SystemId is equal to ***QaStatusPrimaryOrPrimaryBypassSystemId***.
2) Year is equal to the year being checked.
3) Quarter is equal to the quarter being checked.
4) OpSuppDataTypeCode = "OP".

If (***SystemOperatingSuppDataRecordsByLocation*** is found)
Set *OperatingHourCount* = ***SystemOperatingSuppDataRecordsByLocation***.Hours.
Else
Set *OperatingHourCount* = null.

Else
Locate a record in ***OperatingSuppDataRecordsByLocation*** where the reporting period is equal to the year/quarter being checked and the OpTypeCode = "OPHOURS" and FuelCode is null.

If (***OperatingSuppDataRecordsByLocation*** is found)
Set *OperatingHourCount* = ***OperatingSuppDataRecordsByLocation***.OpValue.
Else
Set *OperatingHourCount* = null.

if (*OperatingHourCount* is NOT null)
Add *OperatingHourCount* to *GraceOpHours.*

if (*GraceOpHours* > 168)

Set ***CurrentLinearityStatus*** = "OOC-Expired".
exit for.

else

Set ***CurrentLinearityStatus*** = "Missing Op Data".
Set ***LinearityMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
exit for.

if (***CurrentLinearityStatus*** is null)

Set ***CurrentLinearityStatus*** = "IC-Grace".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2 Linearity Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- Hg Linearity Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- NOX Linearity Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB CO2C, NOXC or O2C Linearity St
5	Process/Category:	Emissions Data Evaluation Report --------- O2 Dry Linearity Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- O2 Wet Linearity Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- SO2 Linearity Status Evaluation

Check Code: LINSTAT-5

Check Name: Determine Event Conditional Status

Related Former Checks:

Applicability: CEM Check

Description: If a QA Cert Event was found that affects this MHV record, evaluate the conditional status.

Specifications:

Set ***SubsequentLinearityRecord*** = null.

if (***CurrentLinearityStatus*** is null and ***PriorLinearityEventRecord*** is not null)

if (***PriorLinearityEventRecord***.ConditionalDataBeginDate/Hour is null or ***CurrentDateHour*** is prior to the ***PriorLinearityEventRecord***.ConditionalDataBeginDate/Hour)

Set ***CurrentLinearityStatus*** = "OOC-Event".

else

Locate the earliest record in ***LinearityTestRecordsByLocationForQAStatus*** where the ComponentID is equal to the ***ApplicableComponentID,*** the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed,*** the CalculatedTestResult is not equal to "INVALID" and the EndDate/Hour is on or after the ***PriorLinearityEventRecord***.ConditionalDataBeginDate/Hour.

if (***LinearityTestRecordsByLocationForQAStatus*** is found)

Set ***SubsequentLinearityRecord*** = the found record in ***LinearityTestRecordsByLocationForQAStatus***.

if (***LinearityTestRecordsByLocationForQAStatus***.QANeedsEvaluationFlag = "Y")

Set ***CurrentLinearityStatus*** = "Recertification Test Not Yet Evaluated".

else if (***LinearityTestRecordsByLocationForQAStatus***.TestResultCode is null)

Set ***CurrentLinearityStatus*** = "OOC-Recertification Test Has Critical Errors".

else if (***LinearityTestRecordsByLocationForQAStatus***.TestResultCode = "FAILED")

Set ***CurrentLinearityStatus*** = "OOC-Recertification Test Failed".

else if (***LinearityTestRecordsByLocationForQAStatus***.TestResultCode = "ABORTED")

Set ***CurrentLinearityStatus*** = "OOC-Recertification Test Aborted".

If (***InvalidLinearityRecord*** is null)

Locate the earliest record in ***LinearityTestRecordsByLocationForQAStatus*** where the ComponentID is equal to the ***ApplicableComponentID,*** the SpanScaleCode is equal to the ***CurrentAnalyzerRangeUsed,*** the CalculatedTestResult is equal to "INVALID" and the EndDate/Hour is on or after the ***PriorLinearityEventRecord***.ConditionalDataBeginDate/Hour and is before the EndDate/EndHour of the ***LinearityTestRecordsByLocationForQAStatus*** record retrieved above.

if (***LinearityTestRecordsByLocationForQAStatus*** is found)
Set ***InvalidLinearityRecord*** = the found record in ***LinearityTestRecordsByLocationForQAStatus***.

if (***PriorLinearityEventRecord***.SystemTypeCode == "HG") AND (***PriorLinearityEventRecord***.EventCode in set {"100",

"101", "120", "125")

Set ***CurrentLinearityStatus*** = "IC-Skip Duplicate Checking for Hg Conditional Data".

if (***CurrentLinearityStatus*** is null AND ***Annual Reporting Requirement*** == false)

If (***SubsequentLinearityRecord*** is not null and ***SubsequentLinearityRecord***.EndDate/Hour is greater than October 30th of the year of the ***CurrentDateHour***) OR (***SubsequentLinearityRecord*** is null and the ***CurrentDateHour*** is in the 3rd quarter))

Set ***CurrentLinearityStatus*** = "OOC-Conditional Period Expired".

if (***CurrentLinearityStatus*** is null)

if (***PriorLinearityEventRecord***.LinearityCertEvent == "Y") and (***PriorLinearityEventRecord***.SystemTypeCode is NOT in set (ST))

if (***PriorLinearityEventRecord***.EventCode = 125)

If (***PriorLinearityEventRecord***.MonitoringSystemID is null)
Set ***CurrentLinearityStatus*** = "Invalid Certification Event"

else if (the associated BeginDate of the system in the ***PriorLinearityEventRecord*** is null)
Set ***CurrentLinearityStatus*** = "Invalid Monitor System"

else
If (the associated SystemTypeCode of the system in the ***PriorLinearityEventRecord*** == "SO2")

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where the ProgramCode is in ***ProgramRequiresSo2SystemCertificationList*** and the UnitMonitorCertBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorLinearityEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)
Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the ProgramCode is in ***ProgramRequiresSo2SystemCertificationList*** and the EmissionsRecordingBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorLinearityEventRecord***.

else if (the associated SystemTypeCode of the system in the ***PriorLinearityEventRecord*** == "NOX")

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where the ProgramCode is in ***ProgramRequiresNoxSystemCertificationList*** and the UnitMonitorCertBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorLinearityEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)
Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the ProgramCode is in ***ProgramRequiresNoxSystemCertificationList*** and the EmissionsRecordingBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorLinearityEventRecord***.

else if (the associated SystemTypeCode of the system in the ***PriorLinearityEventRecord*** == "NOXC")

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where the ProgramCode is in ***ProgramRequiresNoxcSystemCertificationList*** and the UnitMonitorCertBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorLinearityEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)
Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the ProgramCode is in ***ProgramRequiresNoxcSystemCertificationList*** and the EmissionsRecordingBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorLinearityEventRecord***.

else if (the associated SystemTypeCode of the system in the ***PriorLinearityEventRecord*** == "HG")

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where the ProgramCode is in set {MATS} and the UnitMonitorCertBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorLinearityEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)
Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the ProgramCode is in set {MATS} and the EmissionsRecordingBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorLinearityEventRecord***.

else
Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where the UnitMonitorCertBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorLinearityEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)
Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the EmissionsRecordingBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorLinearityEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)
Set ***CurrentLinearityStatus*** = "Missing Program".

else if (***LocationProgramRecordsByHourLocation***UnitMonitorCertDeadline is not null)

if (CurrentDate is prior to the ***LocationProgramRecordsByHourLocation***.UnitMonitorCertDeadline)
Set ***CurrentLinearityStatus*** = "IC-Conditional".

else
Set ***CurrentLinearityStatus*** = "OOC-Conditional Period Expired".

else

if (CurrentDate is prior to the ***LocationProgramRecordsByHourLocation***.UnitMonitorCertBeginDate + 180 days)

Set ***CurrentLinearityStatus*** = "IC-Conditional".

else

Set ***CurrentLinearityStatus*** = "OOC-Conditional Period Expired".

else

If (the number of calendar days ON OR AFTER the ***PriorLinearityEventRecord***.QACertEventDate and ON OR BEFORE the ***CurrentDateHour*** > 180)

Set ***CurrentLinearityStatus*** = "OOC-Conditional Period Expired".

else if (the quarter of the ***PriorLinearityEventRecord***.QACertEventDate is equal to the quarter of the ***CurrentDateHour***)

If (the number of calendar days ON OR AFTER the ***PriorLinearityEventRecord***.QACertEventDate and ON OR BEFORE the ***CurrentDateHour*** > 90)

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusComponentTypeCode*** is “CO2”, “NOX” or “O2”)

If (Days in ***QaCertEventSuppDataDictionaryArray*** for the current location and QA Cert Event Date where QaCertEventKey is equal to ***PriorLinearityEventRecord***.QaCertEventKey > 90)

Set ***CurrentLinearityStatus*** = "OOC-Conditional Period Expired".

Else

Set ***CurrentLinearityStatus*** = "IC-Conditional".

Else

If (***Rpt Period Op Hours Accumulator Array*** for the location == -1)

Set ***CurrentLinearityStatus*** = "Invalid Op Data".

else if (the number of calendar days ON OR AFTER the ***PriorLinearityEventRecord***.QACertEventDate and ON OR BEFORE the ***CurrentDateHour*** is equal to ***Rpt Period Op Days Accumulator Array*** for the location)

Set ***CurrentLinearityStatus*** = "OOC-Conditional Period Expired".

else

Set ***CurrentLinearityStatus*** = "IC-Conditional".

else

Set ***CurrentLinearityStatus*** = "IC-Conditional".

else if (***PriorLinearityEventRecord***.MinOpDaysPriorQuarter is null)

Set ***PriorLinearityEventRecord***.MinOpDaysPriorQuarter = 0
Set ***PriorLinearityEventRecord***.MaxOpDaysPriorQuarter = 0

for each quarter beginning with the quarter of the

PriorLinearityEventRecord.QACertEventDate and continuing through the quarter BEFORE the ***CurrentDateHour***:

if (***EarliestLocationReportDate*** <= the last day of the quarter being checked)

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusComponentTypeCode*** is “CO2”, “NOX” or “O2”)

Locate a record in ***SystemOperatingSuppDataRecordsByLocation*** where:

1) SystemId is equal to ***QaStatusPrimaryOrPrimaryBypassSystemId***.
2) Year is equal to the year being checked.
3) Quarter is equal to the quarter being checked.
4) OpSuppDataTypeCode = "OP".

If (***SystemOperatingSuppDataRecordsByLocation*** is found)
Set *OperatingDayCount* = ***SystemOperatingSuppDataRecordsByLocation***.Days.
Else
Set *OperatingDayCount* = null.

Else

Locate the record in ***OperatingSuppDataRecordsbyLocation*** where the OpTypeCode is equal to "OPDAYS" and the reporting period is equal to the quarter being checked.

If (***OperatingSuppDataRecordsbyLocation*** is found)
Set *OperatingDayCount* = ***OperatingSuppDataRecordsByLocation***.OpValue.
Else
Set *OperatingDayCount* = null.

if (*OperatingDayCount* is null)

Set ***PriorLinearityEventRecord***.MinOpDaysPriorQuarter = -1
Set ***LinearityMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.
exit for.

else if (the quarter being checked is the quarter of the ***PriorLinearityEventRecord***.QACertEventDate)

supplementalCount = null.

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusComponentTypeCode*** is “CO2”, “NOX” or “O2” AND ***PriorLinearityEventRecord***.QaCertEventDateSystemSuppData Exists is true)

supplementalCount = ***PriorLinearityEventRecord***.QaCertEventSystemOpDaysCount.

If (*supplementalCount* is null AND PriorLinearityEventRecord.QaCertEventDateSuppDataExists is true)

supplementalCount = ***PriorLinearityEventRecord***.QaCertEventOpDaysCount .

If (*supplementalCount* is NOT null)

Set ***PriorLinearityEventRecord***.MinOpDaysPriorQuarter = ***PriorLinearityEventRecord***.MinOpDaysPriorQuarter + *supplementalCount*.
Set ***PriorLinearityEventRecord***.MaxOpDaysPriorQuarter = ***PriorLinearityEventRecord***.MaxOpDaysPriorQuarter + *supplementalCount*.

Else

If (*OperatingDayCount* MINUS the number of calendar days in the quarter being checked that are PRIOR to the ***PriorLinearityEventRecord***.QACertEventDate > 0)

Set ***PriorLinearityEventRecord***.MinOpDaysPriorQuarter = *OperatingDayCount* MINUS the number of calendar days in the quarter being checked that are PRIOR to the ***PriorLinearityEventRecord***.QACertEventDate

If (*OperatingDayCount* is less than the number of calendar days in the quarter being checked that are ON OR AFTER the ***PriorLinearityEventRecord***.QACertEventDate)

Set ***PriorLinearityEventRecord***.MaxOpDaysPriorQuarter = *OperatingDayCount*.

else

Set ***PriorLinearityEventRecord***.MaxOpDaysPriorQuarter = the number of calendar days in the quarter being checked that are ON OR AFTER the ***PriorLinearityEventRecord***.QACertEventDate .

else

Set ***PriorLinearityEventRecord***.MinOpDaysPriorQuarter = ***PriorLinearityEventRecord***.MinOpDaysPriorQuarter +

OperatingDayCount.
Set ***PriorLinearityEventRecord***.MaxOpDays PriorQuarter = ***PriorLinearityEventRecord***.MaxOpDaysPriorQuarter + *OperatingDayCount*.

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusComponentTypeCode*** is "CO2", "NOX" or "O2")

Set *CurrentOpDays* to Days in ***SystemOperatingSuppDataDictionaryArray*** for the current location where SystemId is equal to ***QaStatusPrimaryOrPrimaryBypassSystemId***.

Else

Set *CurrentOpDays* to ***Rpt Period Op Days Accumulator Array*** for the Location.

if (***PriorLinearityEventRecord***.MinOpDaysPriorQuarter == -1
set ***CurrentLinearityStatus*** to "Missing Op Data"

else if (***PriorEventRecord***.MinOpDaysPriorQuarter + *CurrentOpDays* > 90)
Set ***CurrentLinearityStatus*** = "OOC-Conditional Period Expired".

else if (***PriorEventRecord***.MinOpDaysPriorQuarter == ***PriorEventRecord***.MaxOpDaysPriorQuarter)
Set CurrentLinearityStatus = "IC-Conditional".

else if (***PriorEventRecord***.MaxOpDaysPriorQuarter + *CurrentOpDays* > 90)
Set ***CurrentLinearityStatus*** = "Undetermined-Conditional Data".

else
Set ***CurrentLinearityStatus*** = "IC-Conditional".

else
Set ***CurrentLinearityStatus*** = "IC-Conditional".

else

If (the quarter of the ***PriorLinearityEventRecord***.ConditionalBeginDate is equal to the quarter of the ***CurrentDateHour***)

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusComponentTypeCode*** is "CO2", "NOX" or "O2")

If (Hours in ***QaCertEventSuppDataDictionaryArray*** for the current location and Conditional Data Begin Hour where QaCertEventKey is equal to ***PriorLinearityEventRecord***.QaCertEventKey > 168)
Set ***CurrentLinearityStatus*** = "OOC-Conditional Period Expired".
Else
Set ***CurrentLinearityStatus*** = "IC-Conditional".

Else

Count the number of ***HourlyOpData*** records for the location where OpTime is greater than 0 and Date/Hour is ON OR AFTER the ***PriorLinearityEventRecord***.ConditionalBeginDate/Hour and ON OR BEFORE the ***CurrentDateHour***,

If the number > 168,
Set ***CurrentLinearityStatus*** = "OOC-Conditional Period Expired".

else
Set ***CurrentLinearityStatus*** = "IC-Conditional".

else

if (***PriorLinearityEventRecord***.MinOpHoursPriorQuarter is null)

Set ***PriorLinearityEventRecord***.MinOpHoursPriorQuarter = 0
Set ***PriorLinearityEventRecord***.MaxOpHoursPriorQuarter = 0

for each quarter beginning with the quarter of the ***PriorLinearityEventRecord***.ConditionalBeginDate and continuing through the quarter BEFORE the ***CurrentDateHour***:

if (***EarliestLocationReportDate*** <= the last day of the quarter being checked)

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusComponentTypeCode*** is “CO2”, “NOX” or “O2”)

Locate a record in ***SystemOperatingSuppDataRecordsByLocation*** where:

1) SystemId is equal to ***QaStatusPrimaryOrPrimaryBypassSystemId***.
2) Year is equal to the year being checked.
3) Quarter is equal to the quarter being checked.
4) OpSuppDataTypeCode = "OP" if ***Annual Reporting Requirement*** == true OR the quarter being checked != 2, otherwise “OPMJ”.

If (***SystemOperatingSuppDataRecordsByLocation*** is found)
Set *OperatingHourCount* = ***SystemOperatingSuppDataRecordsByLocation***.Hours.
Else
Set *OperatingHourCount* = null.

Else

if (***Annual Reporting Requirement*** == false) AND (the quarter being checked == 2)

Locate the record in ***OperatingSuppDataRecordsbyLocation*** where the OpTypeCode is equal to "OSHOURS" and the reporting period is equal to the quarter being checked.

else

Locate the record in ***OperatingSuppDataRecordsbyLocation*** where the OpTypeCode is equal to "OPHOURS", FuelCode is null, and the reporting period is equal to the quarter being checked.

If (***OperatingSuppDataRecordsByLocation*** is found)
Set *OperatingHourCount* = ***OperatingSuppDataRecordsByLocation***.OpValue.
Else
Set *OperatingHourCount* = null.

if (*OperatingHourCount* is null)

Set ***PriorLinearityEventRecord***.MinOpHoursPriorQuarter = -1
Set ***LinearityMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
exit for.

else if (the quarter being checked is the quarter of the ***PriorLinearityEventRecord***.ConditionalBeginDate)

supplementalCount = null.

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusComponentTypeCode*** is “CO2”, “NOX” or “O2” AND ***PriorLinearityEventRecord***.ConditionalBeginHourSystemSuppDataExists is true)

supplementalCount = ***PriorLinearityEventRecord***.ConditionalBeginSystemOpHoursCount.

If (*supplementalCount* is null AND PriorLinearityEventRecord.ConditionalBeginHourSuppDataExists is true)

supplementalCount = ***PriorLinearityEventRecord***.ConditionalBeginOpHoursCount.

If (*supplementalCount* is NOT null)

Set ***PriorLinearityEventRecord***.MinOpHoursPriorQuarter = ***PriorLinearityEventRecord***.MinOpHoursPriorQuarter + *supplementalCount*.
Set ***PriorLinearityEventRecord***.MaxOpHoursPriorQuarter = ***PriorLinearityEventRecord***.MaxOpHoursPriorQuarter + *supplementalCount*.

Else
If (*OperatingHourCount* MINUS the number of calendar hours in the quarter being checked that are PRIOR to the ***PriorLinearityEventRecord***.ConditionalBeginDate/Ho

ur > 0)

Set ***PriorLinearityEventRecord***.MinOpHoursPriorQuarter = *OperatingHourCount* MINUS the number of calendar hours in the quarter being checked that are PRIOR to the ***PriorLinearityEventRecord***.ConditionalBeginDate/Hour

If (*OperatingHourCount* is less than the number of calendar hours in the quarter begin checked that are ON OR AFTER the ***PriorLinearityEventRecord***.ConditionalBeginDate/Hour)

Set ***PriorLinearityEventRecord***.MaxOpHoursPriorQuarter = *OperatingHourCount*.

else

Set ***PriorLinearityEventRecord***.MaxOpHoursPriorQuarter = the number of calendar hours in the quarter being checked that are ON OR AFTER the ***PriorLinearityEventRecord***.ConditionalBeginDate/Hour.

else

Set ***PriorLinearityEventRecord***.MinOpHoursPriorQuarter = ***PriorLinearityEventRecord***.MinOpHoursPriorQuarter + *OperatingHourCount*.
Set ***PriorLinearityEventRecord***.MaxOpHours PriorQuarter = ***PriorLinearityEventRecord***.MaxOpHoursPriorQuarter + *OperatingHourCount*.

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusComponentTypeCode*** is “CO2”, “NOX” or “O2”)

Set *CurrentOpHours* to Hours in ***SystemOperatingSuppDataDictionaryArray*** for the current location where SystemId is equal to ***QaStatusPrimaryOrPrimaryBypassSystemId***.

Else

Set *CurrentOpHours* to ***Rpt Period Op Hours Accumulator Array*** for the Location.

if (***PriorLinearityEventRecord***.MinOpHoursPriorQuarter == -1)
set ***CurrentLinearityStatus*** to "Missing Op Data"

else if (***Rpt Period Op Days Accumulator Array*** for the location == -1)

if (***PriorEventRecord***.MinOpHoursPriorQuarter > 168)
Set ***CurrentLinearityStatus*** = "OOC-Conditional Period Expired".

else
Set ***CurrentLinearityStatus*** = "Invalid Op Data".

else
if (***PriorEventRecord***.MinOpHoursPriorQuarter + *CurrentOpHours* > 168)
Set ***CurrentLinearityStatus*** = "OOC-Conditional Period Expired".

else if (***PriorEventRecord***.MinOpHoursPriorQuarter ==
PriorEventRecord.MaxOpHoursPriorQuarter)
Set CurrentLinearityStatus = "IC-Conditional".

else if (***PriorEventRecord***.MaxOpHoursPriorQuarter + *CurrentOpHours* > 168)
Set ***CurrentLinearityStatus*** = "Undetermined-Conditional Data".

else
Set ***CurrentLinearityStatus*** = "IC-Conditional".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2 Linearity Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- Hg Linearity Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- NOX Linearity Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB CO2C, NOXC or O2C Linearity St
5	Process/Category:	Emissions Data Evaluation Report --------- O2 Dry Linearity Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- O2 Wet Linearity Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- SO2 Linearity Status Evaluation

Check Code: LINSTAT-6

Check Name: Determine Final Linearity Status

Related Former Checks:

Applicability: CEM Check

Description: Evaluates the determined Linearity Status and changes it if needed based on an ignored test or the status of the alternate range.

Specifications:

Set ***AlternateLinearityRecord*** = null
Set ***AlternateInvalidLinearityRecord*** = null

if (***CurrentLinearityStatus*** begins with "OOC")

    if (***InvalidLinearityRecord*** is not null)
        Set ***CurrentLinearityStatus*** = ***CurrentLinearityStatus*** & "*".

    Return result ***CurrentLinearityStatus***.

else if (***CurrentLinearityStatus***= "Invalid Monitor Span")

    if (***CurrentAnalyzerRangeUsed*** = "H")

        Set ***CurrentLinearityStatus*** = ***CurrentLinearityStatus*** & " (High Scale)".

    else

        Set ***CurrentLinearityStatus*** = ***CurrentLinearityStatus*** & " (Low Scale)".

    Return result ***CurrentLinearityStatus***.

else if (***DualRangeStatus*** = true and ***CurrentLinearityStatus*** begins with "IC" or "Undetermined")

    if (***CurrentAnalyzerRangeUsed*** = "H")

        Set *AlternateAnalyzerRange* = "L".
        Set *AlternateComponentID* = ***LowRangeComponentID***.

    else

        Set *AlternateAnalyzerRange* = "H".
        Set *AlternateComponentID* = ***HighRangeComponentID***.

    for each record in ***MonitorSystemComponentRecordsByHourLocation*** where the ComponentID is equal to the *AlternateComponentID*

        Append ***MonitorSystemComponentRecordsByHourLocation***.SystemID to *AlternateSystemIDs*.

    if (***MonitorSystemComponentRecordsByHourLocation*** is not found)

        Set ***CurrentLinearityStatus*** = "Invalid Monitor System Component".
        Return result ***CurrentLinearityStatus***.

    if (***CurrentMHVParameter*** in set {SO2C, NOXC})

        Locate the record in ***MonitorSpanRecordsByHourAndLocation*** for the hour where the ComponentTypeCode is equal to the ***QaStatusComponentTypeCode*** and the SpanScaleCode is equal to the *AlternateAnalyzerRange*.

if (***MonitorSpanRecordsByHourAndLocation*** is not found OR more than one ***MonitorSpanRecordsByHourAndLocation*** is found or ***MonitorSpanRecordsByHourAndLocation***.SpanValue is null or is less than or equal to 0)

Set ***CurrentLinearityStatus*** = "Invalid Monitor Span".

if (*AlternateAnalyzerRange* = "H")

Set ***CurrentLinearityStatus*** = ***CurrentLinearityStatus*** & " (High Scale)".

else

Set ***CurrentLinearityStatus*** = ***CurrentLinearityStatus*** & " (Low Scale)".

Return result ***CurrentLinearityStatus***.

else if (***MonitorSpanRecordsByHourAndLocation***.SpanValue < =30)
If (***CurrentLinearityStatus*** does not begin with "IC")
Return result ***CurrentLinearityStatus***.
else
exit check.

Locate the most recent record in ***LinearityTestRecordsByLocationForQAStatus*** for the *AlternateComponentID* where the SpanScaleCode is equal to the *AlternateAnalyzerRange* and the CalculatedTestResult is not equal to "INVALID" and the EndDate/Hour is either:

a) prior to the ***CurrentDateHour*** OR
b) equal to the ***CurrentDateHour*** and the EndMinute is less than "45" and the CalculatedTestResult is equal to "PASSED" or "PASSAPS".

if (***LinearityTestRecordsByLocationForQAStatus*** is found)
Set ***AlternateLinearityRecord*** = the found record in ***LinearityTestRecordsByLocationForQAStatus***.

Locate all records in ***QACertificationEventRecords*** where:

the ComponentID is equal to the *AlternateComponentID*

AND LinearityRequired is equal to "Y",

AND the QACertEventDate/Hour is either:

a) prior to the ***CurrentDateHour*** OR
b) equal to both the ***CurrentDateHour*** and the ConditionalDataBeginDate/Hour;

AND either

a) ***AlternateLinearityRecord*** is null OR
b) QACertEventDate/Hour is after the ***AlternateLinearityRecord***.EndDate/Hour OR
c) QACertEventDate/Hour is equal to the ***AlternateLinearityRecord***.EndDate/Hour AND (TestCompletionDate is null or the TestCompletionDate/Hour is after the ***AlternateLinearityRecord***.EndDate/Hour)

AND either

a) ***DualRangeStatus*** = false OR
b) ***HighRangeComponentID*** <> ***LowRangeComponentID*** OR
c) QACertEventCode <> 27 or 30 or 172 and *AlternateAnalyzerRange* = "H" OR
d) QACertEventCode <> 35 or 171 and *AlternateAnalyzerRange* = "L"

if (***QACertificationEventRecords*** is found)

if (***QACertificationEventRecords***.ConditionalBeginDate/Hour is null or ***CurrentDateHour*** is prior to the ***QACertificationEventRecords***.ConditionalBeginDate/Hour)
If (***CurrentLinearityStatus*** does not begin with "IC")
Return result ***CurrentLinearityStatus***.

else
Locate the most recent record in ***LinearityTestRecordsByLocationForQAStatus*** for the *AlternateComponentID* where the SpanScaleCode is equal to the *AlternateAnalyzerRange,* the CalculatedTestResult is equal to "INVALID", and the EndDate/Hour is after the ***QACertificationEventRecords***.QACertEventDate/Hour and prior to the ***CurrentDateHour***.

if (***LinearityTestRecordsByLocationForQAStatus*** is found)
Set ***AlternateInvalidLinearityRecord*** = the found record in ***LinearityTestRecordsByLocationForQAStatus***.

else
Set ***AlternateInvalidLinearityRecord*** = null.

Locate the first record in ***LinearityTestRecordsByLocationForQAStatus*** where the ComponentID is equal to the *AlternateComponentID,* the SpanScaleCode is equal to the *AlternateAnalyzerRange,*the CalculatedTestResult is not equal to "INVALID", and the EndDate/Hour is on or after the ***QACertificationEventRecords***.ConditionalDataBeginDate/Hour.

if (***LinearityTestRecordsByLocationForQAStatus*** is found)
Set ***AlternateLinearityRecord*** = the found record in ***LinearityTestRecordsByLocationForQAStatus***.

if (***AlternateLinearityRecord*** .QANeedsEvaluationFlag = "Y")
Set ***CurrentLinearityStatus*** = "Alternate Range Recertification Test Not Yet Evaluated".

else if (***AlternateLinearityRecord*** .TestResultCode is null or is in set {FAILED, ABORTED})

If (***AlternateInvalidLinearityRecord*** is null)

Locate the most recent record in ***LinearityTestRecordsByLocationForQAStatus*** for the *AlternateComponentID* where the SpanScaleCode is equal to the *AlternateAnalyzerRange,* the CalculatedTestResult is equal to "INVALID", and the EndDate/Hour is after the ***QACertificationEventRecords***.QACertEventDate/Hour and prior to the EndDate/Hour of the ***LinearityTestRecordsByLocationForQAStatus*** record retrieved above.

if (***LinearityTestRecordsByLocationForQAStatus*** is found)
Set ***AlternateInvalidLinearityRecord*** = the found record in ***LinearityTestRecordsByLocationForQAStatus***.

if (***AlternateLinearityRecord*** .TestResultCode is null)
Set ***CurrentLinearityStatus*** = "OOC-Alternate Range Recertification Test Has Critical Errors".
if (***AlternateInvalidLinearityRecord*** is not null)
Set ***CurrentLinearityStatus*** = ***CurrentLinearityStatus*** & "*".

else if (***AlternateLinearityRecord*** .TestResultCode = "FAILED")
Set ***CurrentLinearityStatus*** = "OOC-Alternate Range Recertification Test Failed".
if (***AlternateInvalidLinearityRecord*** is not null)
Set ***CurrentLinearityStatus*** = ***CurrentLinearityStatus*** & "*".

else if (***AlternateLinearityRecord*** .TestResultCode = "ABORTED")
Set ***CurrentLinearityStatus*** = "OOC-Alternate Range Recertification Test Aborted".
if (***AlternateInvalidLinearityRecord*** is not null)
Set ***CurrentLinearityStatus*** = ***CurrentLinearityStatus*** & "*".

If (***CurrentLinearityStatus*** does not begin with "IC")
Return result ***CurrentLinearityStatus***.

else
if (***AlternateLinearityRecord*** is found)
if (***AlternateLinearityRecord***.QANeedsEvaluationFlag = "Y")
Set ***CurrentLinearityStatus*** = "Alternate Range Test Not Yet Evaluated".

else if (***AlternateLinearityRecord.TestResultCode*** is null or is in set {ABORTED, FAILED})

Locate the most recent record in ***LinearityTestRecordsByLocationForQAStatus*** for the *AlternateComponentID* where the SpanScaleCode is equal to the *AlternateAnalyzerRange* and the EndDate/Hour is prior to the ***CurrentDateHour*** and the EndDate/Hour is greater than the ***AlternateLinearityRecord***.EndDate/Hour and the CalculatedTestResult is equal to "INVALID".

if (***LinearityTestRecordsByLocationForQAStatus*** is found)
Set ***AlternateInvalidLinearityRecord*** = the found record in ***LinearityTestRecordsByLocationForQAStatus***.

if (***AlternateLinearityRecord*** .TestResultCode = null)
Set ***CurrentLinearityStatus*** = "OOC-Alternate Range Test Has Critical Errors".
if (***AlternateInvalidLinearityRecord*** is not null)
Set ***CurrentLinearityStatus*** = ***CurrentLinearityStatus*** & "*".

else if (***AlternateLinearityRecord*** .TestResultCode = "FAILED")
Set ***CurrentLinearityStatus*** = "OOC-Alternate Range Test Failed".
if (***AlternateInvalidLinearityRecord*** is not null)
Set ***CurrentLinearityStatus*** = ***CurrentLinearityStatus*** & "*".

else if (***AlternateLinearityRecord*** .TestResultCode = "ABORTED")
Set ***CurrentLinearityStatus*** = "OOC-Alternate Range Test Aborted".
if (***AlternateInvalidLinearityRecord*** is not null)
Set ***CurrentLinearityStatus*** = ***CurrentLinearityStatus*** & "*".
else
Set ***CurrentLinearityStatus*** = "OOC-No Prior Alternate Range Test or Event".

If (***CurrentLinearityStatus*** does not begin with "IC")
Return result ***CurrentLinearityStatus***.
else
If (***CurrentLinearityStatus*** does not begin with "IC")
Return result ***CurrentLinearityStatus***.

Results:

Result	Response	Severity
Alternate Range Recertification Test Not Yet Evaluated	The [testtype] status for [key] could not be determined, because the prior [testtype] for the alternate range component with TestNumber [alttestnum] has not yet been evaluated.	Critical Error Level 1
Alternate Range Test Not Yet Evaluated	The [testtype] status for [key] could not be determined, because the prior [testtype] for the alternate range component with TestNumber [alttestnum] has not yet been evaluated.	Critical Error Level 1
Invalid Certification Event	The [testtype] status for [key] could not be determined, because the QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] has a critical error.	Critical Error Level 1
Invalid Monitor Span (High Scale)	The [testtype] status for [key] could not be determined, because you did not report a single, valid high-scale [comptype] span record that was active during the test.	Critical Error Level 1
Invalid Monitor Span (Low Scale)	The [testtype] status for [key] could not be determined, because you did not report a single, valid low-scale [comptype] span record that was active during the test.	Critical Error Level 1
Invalid Monitor System	The [testtype] status for [key] could not be determined, because the Monitor System record for MonitoringSystemID [system] has a critical error.	Critical Error Level 1
Invalid Monitor System Component	The [testtype] status for [key] could not be determined, because you did not report any active Monitor System Component records for the alternate range of the component.	Critical Error Level 1
Invalid Op Data	The [testtype] status for [key] could not be determined, because the OperatingTime in at least one Hourly Operating Data records was missing or invalid.	Critical Error Level 1
Missing Op Data	The [testtype] status for [key] could not be determined, because the Op Supp Data record for OPHOURS, OSHOURS, or OPDAYS is missing for [MISSINGOPDATAINFO] (and possibly other previous reporting periods). If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
Missing Program	The [testtype] status for [key] could not be determined, because a Unit Program record associated with the initial certification event for QACertEventCode [code] QACertEventDate [eventdate] either does not exist or has a UnitMonitorCertificationBeginDate inconsistent with the BeginDate of the associated Monitor System record.	Critical Error Level 1
OOC-Alternate Range Recertification Test Aborted	The subsequent recertification [testtype] for the alternate range of the component for [key] with TestNumber [alttestnum] was aborted.	Critical Error Level 1
OOC-Alternate Range Recertification Test Aborted*	The subsequent recertification [testtype] for the alternate range of [key] with TestNumber [alttestnum] was aborted. An invalid [testtype] with TestNumber [altinvtestnum] was ignored.	Critical Error Level 1
OOC-Alternate Range Recertification Test Failed	The subsequent recertification [testtype] for the alternate range of the component for [key] with TestNumber [alttestnum] failed.	Critical Error Level 1
OOC-Alternate Range Recertification Test Failed*	The subsequent recertification [testtype] for the alternate range of the component for [key] with TestNumber [alttestnum] failed. An invalid [testtype] with TestNumber [altinvtestnum] was ignored.	Critical Error Level 1
OOC-Alternate Range Recertification Test Has Critical Errors	The subsequent recertification [testtype] for the alternate range of the component for [key] with TestNumber [alttestnum] has critical errors.	Critical Error Level 1

OOC-Alternate Range Recertification Test Has Critical Errors*	The subsequent recertification [testtype] for the alternate range of the component for [key] with TestNumber [alttestnum] has critical errors. An invalid [testtype] with TestNumber [altinvtestnum] was ignored.	Critical Error Level 1
OOC-Alternate Range Test Aborted	The prior [testtype] for the alternate range of the component for [key] with TestNumber [alttestnum] was aborted.	Critical Error Level 1
OOC-Alternate Range Test Aborted*	The prior [testtype] for the alternate range of the component for [key] with TestNumber [alttestnum] was aborted. An invalid [testtype] with TestNumber [altinvtestnum] was ignored.	Critical Error Level 1
OOC-Alternate Range Test Failed	The prior [testtype] for the alternate range of the component for [key] with TestNumber [alttestnum] failed.	Critical Error Level 1
OOC-Alternate Range Test Failed*	The prior [testtype] for the alternate range of the component for [key] with TestNumber [alttestnum] failed. An invalid [testtype] with TestNumber [altinvtestnum] was ignored.	Critical Error Level 1
OOC-Alternate Range Test Has Critical Errors	The prior [testtype] for the alternate range of the component for [key] with TestNumber [alttestnum] has critical errors.	Critical Error Level 1
OOC-Alternate Range Test Has Critical Errors*	The prior [testtype] for the alternate range of the component for [key] with TestNumber [alttestnum] has critical errors. An invalid [testtype] with TestNumber [altinvtestnum] was ignored.	Critical Error Level 1
OOC-Conditional Period Expired	The conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [key] has expired.	Critical Error Level 1
OOC-Conditional Period Expired*	The conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [key] has expired. A prior test was ignored.	Critical Error Level 1
OOC-Event	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate], but you did not indicate the use of conditional data for [key].	Critical Error Level 1
OOC-Event*	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for [key], but you did not indicate the use of conditional data. An invalid [testtype] was ignored.	Critical Error Level 1
OOC-Expired	The prior [testtype] for [key] with TestNumber [testnum] has expired.	Critical Error Level 1
OOC-Expired*	The prior [testtype] for [key] with TestNumber [testnum] has expired. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-No Prior 3-Point SI or Event	You did not report a prior three-point Hg system integrity check or certification event for [key].	Critical Error Level 1
OOC-No Prior Alternate Range Test or Event	You did not report a prior [testtype] or certification event for the alternate range of the component for [key].	Critical Error Level 1
OOC-No Prior Test or Event	You did not report a prior [testtype] or certification event for [key].	Critical Error Level 1
OOC-No Prior Test or Event*	You did not report a valid prior [testtype] or certification event for [key]. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Recertification Test Aborted	The subsequent recertification [testtype] for [key] with TestNumber [subtestnum] was aborted.	Critical Error Level 1
OOC-Recertification Test Aborted*	The subsequent recertification [testtype] for [key] with TestNumber [subtestnum] was aborted. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Recertification Test Failed	The subsequent recertification [testtype] for [key] with TestNumber [subtestnum] failed.	Critical Error Level 1
OOC-Recertification Test Failed*	The subsequent recertification [testtype] for [key] with TestNumber [subtestnum] failed. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Recertification Test Has Critical Errors	The subsequent recertification [testtype] for [key] with TestNumber [subtestnum] has critical errors.	Critical Error Level 1
OOC-Recertification Test Has Critical Errors*	The subsequent recertification [testtype] for [key] with TestNumber [subtestnum] has critical errors. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1

OOC-Test Aborted	The applicable prior [testtype] for [key] with TestNumber [testnum] was aborted.	Critical Error Level 1
OOC-Test Aborted*	The prior [testtype] for [key] with TestNumber [testnum] was aborted. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Test Failed	The applicable prior [testtype] for [key] with TestNumber [testnum] failed.	Critical Error Level 1
OOC-Test Failed*	The prior [testtype] for [key] with TestNumber [testnum] failed. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Test Has Critical Errors	The applicable prior [testtype] for [key] with TestNumber [testnum] has critical errors.	Critical Error Level 1
OOC-Test Has Critical Errors*	The prior [testtype] for [key] with TestNumber [testnum] has critical errors. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
Prior Test Not Yet Evaluated	The [testtype] status for [key] could not be determined, because the applicable prior [testtype] with TestNumber [testnum] has not yet been evaluated.	Critical Error Level 1
Recertification Test Not Yet Evaluated	The [testtype] status for [key] could not be determined, because the subsequent recertification [testtype] for the component with TestNumber [subtestnum] has not yet been evaluated.	Critical Error Level 1
Undetermined-Conditional Data	The software could not determine if the current hour was within the conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [key]	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2 Linearity Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- Hg Linearity Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- NOX Linearity Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB CO2C, NOXC or O2C Linearity St
5	Process/Category:	Emissions Data Evaluation Report --------- O2 Dry Linearity Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- O2 Wet Linearity Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- SO2 Linearity Status Evaluation

Check Code: LINSTAT-7

Check Name: Ensure Certifiying Three Level System Integrity Test Exists for Component

Related Former Checks:

Applicability: General Check

Description: For a Hg CEMS component, when LINSTAT-4 has assigned a status of IC, IC-Extension or IC-Grace, perform the following:

1) Locate the most recent 3-Level SI.
2) If one does not exist, return a result.
3) Otherwise, locate an intervening certification event (120, 125) for the component.
4) If an event was located, do nothing.
5) Otherwise, return "OOC- No Priior HGSI3 Test or Event".

Specifications:

Set ***MatsCheckForHgsi3Ran*** to false.

If (***CurrentLinearityStatus*** is equal to "IC", "IC-Extension" or "IC-Grace")

If (***PriorLinearityRecord***.ComponentTypeCode is equal to "HG")

Locate the most recent record in ***LinearityTestRecordsByLocationForQAStatus*** where:

1) ComponentID is equal to ***PriorLinearityRecord***.ComponentId.
2) TestTypeCode is equal to "HGSI3".
3) TestResultCode is equal to "PASSED" or "PASSAPS".
4) EndDateHour is prior to ***CurrentDateHour***.

If NOT found,

Set ***CurrentLinearityStatus*** to "OOC-No Prior 3-Point SI or Event".

Else

Count records in ***QACertificationEventRecords*** where:

1) ComponentID is equal to ***PriorLinearityRecord***.ComponentId.
2) QACertEventCode is equal to 120 or 125.
3) QACertEventDate/Hour is prior to ***CurrentDateHour***.
4) QACertEventDate/Hour is after the EndDateHour of the located ***LinearityTestRecordsByLocationForQAStatus*** record.

If the count is greater than 0,

Set ***CurrentLinearityStatus*** to "OOC-No Prior 3-Point SI or Event".

Set ***MatsCheckForHgsi3Ran*** to true.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Hg Linearity Status Evaluation

Check Category:

LME

Check Code: LME-11

Check Name: Check LTFF System

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LTFF record:

If MonitoringSystemID is null,
return result A.

else
Locate the Monitor System record for the MonitoringSystemID.

If the associated SystemTypeCode is not equal to "LTOL" or "LTGS",
return result B.

Results:

Result	Response	Severity
A	You did not report a MonitoringSystemID in an LTFF record.	Fatal
B	The MonitoringSystemID reported in the LTFF record for [key] is not a long-term fuel flow system.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Long Term Fuel Flow
1	Process/Category:	LME Emissions Data Generation LTFF Heat Input Data
1	Process/Category:	Emissions Data Entry Screen Evaluation LTFF Data Evaluation

Check Code: LME-13

Check Name: Check Long Term Fuel Flow Value

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LTFF record:

If the LongTermFuelFlowValue is null or is less than or equal to 0,
return result A.

Results:

Result	Response	Severity
A	The [fieldname] reported in the LTFF record for [key] is missing or invalid.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Long Term Fuel Flow
1	Process/Category:	LME Emissions Data Generation LTFF Heat Input Data
1	Process/Category:	Emissions Data Entry Screen Evaluation LTFF Data Evaluation

Check Code: LME-14

Check Name: Check Long Term Fuel Flow UOM

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LTFF record:

If the LongTermFuelFlowUOMCode is null,
return result A.

Otherwise,
If the SystemTypeCode is "LTOL" and the LongTermFuelFlowUOMCode is not in set {"LB", "GAL"},
return result A.

If the SystemTypeCode is "LTGS" and the LongTermFuelFlowUOMCode is not equal to "SCF",
return result A.

Results:

Result	Response	Severity
A	The [fieldname] reported in the LTFF record for [key] is missing or invalid.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Long Term Fuel Flow
1	Process/Category:	LME Emissions Data Generation LTFF Heat Input Data
1	Process/Category:	Emissions Data Entry Screen Evaluation LTFF Data Evaluation

Check Code: LME-15

Check Name: Check LTFF GCV

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LTFF record:

If the GrossCalorificValue is null or is less than or equal to 0,
return result A.

Results:

Result	Response	Severity
A	The [fieldname] reported in the LTFF record for [key] is missing or invalid.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Long Term Fuel Flow
1	Process/Category:	LME Emissions Data Generation LTFF Heat Input Data
1	Process/Category:	Emissions Data Entry Screen Evaluation LTFF Data Evaluation

Check Code: LME-16

Check Name: Check LTFF GCV UOM

Related Former Checks:

Applicability: LME Check

Description:

Validation Tables:

Fuel Type Reality Checks for GCV (Cross Check Table)
Fuel Type Warning Levels for GCV (Cross Check Table)
Fuel Type Reality Checks for GCV (Cross Check Table)
Fuel Type Warning Levels for GCV (Cross Check Table)
Fuel Type Reality Checks for GCV (Cross Check Table)
Fuel Type Warning Levels for GCV (Cross Check Table)

Specifications:

For the LTFF record:

LME Gen LTFF Heat Input = null.

If the GCVUnitsOfMeasureCode is null,
return result A.

Otherwise,
If the LongTermFuelFlowUOMCode is "LB" and the GCVUnitsOfMeasureCode is not equal to "BTULB",
return result A.

If the LongTermFuelFlowUOMCode is "GAL" and the GCVUnitsOfMeasureCode is not equal to "BTUGAL",
return result A.

If the LongTermFuelFlowUOMCode is "SCF" and the GCVUnitsOfMeasureCode is not equal to "BTUSCF",
return result A.

Otherwise,
If GrossCalorificValue is greater than 0 and LongTermFuelFlowValue is greater than 0,
Calculate ***LME Gen LTFF Heat Input*** = GrossCalorificValue * LongTermFuelFlowValue / 1,000,000, rounded to the nearest integer.

Max Expected GCV = Lookup "Upper Value" in "Fuel Type Warning Levels for GCV Cross Check Table" where "Fuel Code - Units Of Measure" column = concatenation of (FuelCode, " - ", LongTermFuelFlowUOMCode)
Min Expected GCV = Lookup "Lower Value" in "Fuel Type Warning Levels for GCV Cross Check Table" where "Fuel Code - Units Of Measure" column = concatenation of (FuelCode, " - ", LongTermFuelFlowUOMCode)
Max Allowed GCV = Lookup "Upper Value" in "Fuel Type Reality Checks for GCV Cross Check Table" where "Fuel Code - Units Of Measure" column = concatenation of (FuelCode, " - ", LongTermFuelFlowUOMCode)
Min Allowed GCV = Lookup "Lower Value" in "Fuel Type Reality Checks for GCV Cross Check Table" where "Fuel Code - Units Of Measure" column = concatenation of (FuelCode, " - ", LongTermFuelFlowUOMCode)

if (*Max Allowed GCV* is not null AND GrossCalorificValue > *Max Allowed GCV)* OR (*Min Allowed GCV* is not null AND GrossCalorificValue < *Min Allowed GCV*)
return result B
else
if (*Min Expected GCV* is not null AND GrossCalorificValue < *Min Expected GCV)* OR (*Max Expected GCV* is not null AND GrossCalorificValue > *Max Expected GCV*)
return result C

Results:

Result	Response	Severity
A	The [fieldname] reported in the LTFF record for [key] is missing or invalid.	Critical Error Level 1
B	The GrossCalorificValue reported in the long-term fuel flow record for [key] is outside the range of allowable values for the fuel type [fuelcd].	Critical Error Level 1
C	The GrossCalorificValue reported in the long-term fuel flow record for [key] is outside the range of expected values for the fuel type [fuelcd].	Non-Critical Error

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Long Term Fuel Flow
1	Process/Category:	LME Emissions Data Generation LTFF Heat Input Data
1	Process/Category:	Emissions Data Entry Screen Evaluation LTFF Data Evaluation

Check Code: LME-41

Check Name: Check LTFF Fuel Flow Period Code

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LTFF record:

If ***LME OS*** is equal to true,
    If the Quarter of the ***Current Reporting Period*** is equal to 2,
        If the FuelFlowPeriodCode is null,
            return result A.

    Otherwise,
        If the FuelFlowPeriodCode is not null,
            return result B.

Otherwise,
    If the FuelFlowPeriodCode is not null,
        return result C.

Results:

Result	Response	Severity
A	You did not report a FuelFlowPeriodCode in the LTFF record for [key]. This value is required for LME units with an ozone-season qualification during the second quarter.	Critical Error Level 1
B	You reported a FuelFlowPeriodCode in the LTFF record for [key], but this value is only appropriate during the second quarter.	Critical Error Level 1
C	You reported a FuelFlowPeriodCode in the LTFF record for [key], but this value is only appropriate for LME units with an ozone-season qualification.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Long Term Fuel Flow

Check Code: LME-42

Check Name: Check LTFF Total Heat Input

Related Former Checks:

Applicability: LME Check

Description:

Validation Tables:

Hourly Emissions Tolerances (Cross Check Table)

Specifications:

For the Current LTFF record:

If ***LME Gen LTFF Heat Input*** is not null,
If ***LME Total Heat Input Array*** for the location is greater than or equal to 0,
add ***LME Gen LTFF Heat Input*** to ***LME Total Heat Input Array*** for the location.
If ***LME OS*** is true and the ***Current LTFF Record***.FuelFlowPeriodCode is equal to "A",
add ***LME Gen LTFF Heat Input*** to ***LME April Total Heat Input Array*** for the location.

If (the LocationName begins with "CP")
If ***LME CP Total Heat Input*** is greater than or equal to 0,
add ***LME Gen LTFF Heat Input*** to ***LME CP Total Heat Input***.
If ***LME OS*** is true and the ***Current LTFF Record***.FuelFlowPeriodCode is equal to "A",
add ***LME Gen LTFF Heat Input*** to ***LME CP April Heat Input***.

else
Set ***LME Total Heat Input Array*** for the location to -1.

If (the LocationName begins with "CP")
set ***LME CP Total Heat Input*** to -1.

If (the LocationName begins with "CP")
Rpt Period HI Calculated Accumulator Array for this location = ***LME Total Heat Input Array*** for the location.
April HI Calculated Accumulator Array for this location = ***LME Total April Input Array*** for the location
Expected Summary Value HI Array for this location = true

If ***Current LTFF Record***.TotalHeatInput is greater than or equal to 0,

If (the LocationName begins with "CP" AND ***Rpt Period HI Reported Accumulator Array*** for this location is greater than or equal to 0)
Rpt Period HI Reported Accumulator Array for this location = ***Rpt Period HI Reported Accumulator Array*** for this location + ***Current LTFF Record***.TotalHeatInput

If ***LME Gen LTFF Heat Input*** is not null AND ***Current LTFF Record***.TotalHeatInput is not equal to ***LME Gen LTFF Heat Input***,
Heat Input Tolerance = Lookup Tolerance from Cross-Check Table "Hourly Emissions Tolerances" where
Parameter = "HI" AND
UOM = "MMBTUHR"

if (ABS(***Current LTFF Record***.TotalHeatInput - ***LME Gen LTFF Heat Input***) > *Heat Input Tolerance*)
return result A.

else
If (the LocationName begins with "CP")
Rpt Period HI Reported Accumulator Array for this location = -1.

return result B

Results:

Result	Response	Severity
A	The TotalHeatInput reported in the LTFF record for [key] is inconsistent with the recalculated value.	Critical Error Level 1
B	The [fieldname] reported in the LTFF record for [key] is missing or invalid.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- Long Term Fuel Flow

Check Code: LME-10

Check Name: Determine Total Load for Reporting Period

Related Former Checks: LME-EXP8A

Applicability: LME Check

Description:

Specifications:

Set ***LME Gen LTFF Heat Input Array***, ***LME Gen Total Heat Input Array***, ***LME Gen Total Load Array***, ***LME Gen Total SO2M Array, LME Gen Total NOXM Array***, ***LME Gen Total CO2M Array, LME Gen Total Op Time Array***, ***LME Gen Total Op Hours Array, and LME Gen LTFF Total Op Time Array*** to 0 for each location in the monitor configuration.
Set ***LME Gen LTFF April Heat Input Array, LME Gen April Heat Input Array***, ***LME Gen April Load Array, LME Gen April NOXM Array***, ***LME Gen April Op Time Array***, ***LME Gen April Op Hours Array, and LME Gen LTFF April Op Time Array*** to 0 for each location in the monitoring configuration.
Set ***LME Gen CP Total Heat Input, LME Gen Total Load***, and ***LME Gen Total Optime*** to 0.
Set ***LME Gen CP April Heat Input, LME Gen April Load***, and ***LME Gen April Optime*** to 0.
Set ***LME Gen Annual*** and ***LME Gen OS*** to false.
Set ***LME Gen HI Method*** and ***LME Gen HI Substitute Data Code*** to null.

Locate MonitorMethod records for all locations in the monitoring configuration where ParameterCode = "HIT", BeginDate is on or before the first day of the reporting period, and the EndDate is null or is on or after the last day of the reporting period.

If any location does not have a retrieved record,
return result A.

Otherwise,

Set ***LME Year Start Quarter*** to the quarter of the current reporting period.

Locate a MonitorQualification for all units in the monitoring configuration where the QualificationTypeCode is equal to "LMEA" or "LMES", BeginDate is on or before the last day of the reporting period, and the EndDate is null or is on or after January 1 of the year of the reporting period.

If a record with QualificationTypeCode equal to "LMEA" is found,
set ***LME Gen Annual*** to true.

If a record with QualificationTypeCode equal to "LMES" is found,
set ***LME Gen OS*** to true.

If ***LME Gen Annual*** AND ***LME Gen OS*** are both false,
return result B.

else if ***LME Gen Annual*** is false AND the Quarter of the reporting period is equal to 1 or 4,
return result C.

Otherwise,

If the Quarter of the reporting period is greater than 1,

If ***LME Gen Annual*** is equal to true,
set ***LME Year Start Quarter*** to 1.
else
set ***LME Year Start Quarter*** to 2.

If MethodCode in all the retrieved Method records is equal to "MHHI",
LME Gen HI Method = "MHHI".

Locate an LTFF record for any location during the reporting period.

If found,
return result D.

If MethodCode in all the retrieved Method records is in set {LTFF, CALC, LTFCALC},

LME Gen HI Method = "LTFF".

If SubstituteDataCode in any retrieved record is equal to "MHHI",
LME Gen HI Substitute Data Code = "MHHI".

For each ***Hourly Op Data*** record for the configuration:

If ***Hourly Op Data.***HourLoad is not null and is less than 0,
return result E.

else if ***Hourly Op Data.***OpTime for any hour is null, less than 0, or greater than 1,
return result F.

else if ***Hourly Op Data.***OpTime is greater than 0 and ***Hourly Op Data.***HourLoad is null,
return result E.

else if ***Hourly Op Data.***OpTime is greater than 0 AND ***Hourly Op Data.***MHHI Indicator is not equal to 1,
Add HourLoad * OpTime to ***LME Gen Total Load Array*** for the location.
Add HourLoad * OpTime to ***LME Gen Total Load***.
Add OpTime to ***LME Gen Total Optime***.
Add OpTime to ***LME Gen LTFF Total Op Time Array*** for location.

If the month of ***Hourly Op Data.***Date is April AND ***LME Gen OS*** is equal to true,
Add HourLoad * OpTime to ***LME Gen April Load Array*** for the location.
Add HourLoad * OpTime to ***LME Gen April Load***.
Add OpTime to ***LME Gen April Optime***.
Add OpTime to ***LME Gen LTFF April Op Time Array*** for location

If ***LME Gen OS*** is equal to true and the Quarter of the reporting period is equal to 2,

Locate an LTFF record for any location in the monitoring configuration during the reporting period where the FuelFlowPeriodCode is equal to "A".

If found and the ***LME Gen April Load*** is equal to 0 and ***LME Gen April Optime*** is equal to 0,
return result J.

Else if not found AND (***LME Gen April Load*** is greater than 0 or ***LME Gen April Optime*** is greater than 1),
return result K.

Otherwise,

Locate an LTFF record for any location in the monitoring configuration during the reporting period where the FuelFlowPeriodCode is equal to "MJ".

If found,

If (***LME Gen Total Load - LME Gen April Load***) is equal to 0 and (***LME Gen Total Optime - LME Gen April Optime***) is equal to 0,

return result L.

If not found,

If (***LME Gen Total Load - LME Gen April Load***) is greater than 0 or (***LME Gen Total Optime - LME Gen April Optime***) is greater than 1,

return result M.

Otherwise,

Locate an LTFF record for any location in the monitoring configuration during the reporting period.

If found,

If ***LME Gen Total Load*** is equal to 0 and ***LME Gen Total Optime*** is equal to 0,

return result G.

If not found,

If ***LME Gen Total Load*** is greater than 0 or ***LME Gen Total Optime*** is greater than 1,

return result I.

Otherwise,

return result H.

Do not process remaining categories if fatal error is returned.

Results:

Result	Response	Severity
A	You have not reported an active HIT method in your monitoring plan for at least one monitoring location in the configuration.	Fatal
B	You have not reported an active LMEA or LMES qualification record for this configuration in your monitoring plan.	Fatal
C	You have not reported an active LMEA qualification record for this configuration in your monitoring plan, but the reporting period is the first or fourth quarter. Only annual LME units should report in the first or fourth quarter.	Fatal
D	You have reported MHHI as the heat input method for this configuration, but you have reported a long-term fuel flow record.	Fatal
E	You have reported LTFF as the heat input method for this configuration, but the LoadValue in at least one hourly record is missing or invalid.	Fatal
F	You have reported LTFF as the heat input method for this configuration, but the OperatingTime in at least one hourly record is missing or invalid.	Fatal
G	You have reported a long-term fuel flow record for this reporting period, but the sum of the load and operating time values in the hourly records (where MHHIIndicator is not equal to 1) are equal to 0.	Fatal
H	You have not reported the same heat input method in your monitoring plan for all locations in the configuration during the reporting period.	Fatal
I	You have reported LTFF as the heat input method for this configuration, but you have not reported a long-term fuel flow record for this reporting period.	Critical Error Level 1
J	You have reported a long-term fuel flow record for April, but the sum of the load and operating time values in the hourly records (where MHHIIndicator is not equal to 1) are equal to 0.	Fatal
K	You have reported LTFF as the heat input method for this ozone-season reporting configuration, but you have not reported a long-term fuel flow record for April.	Critical Error Level 1
L	You have reported a long-term fuel flow record for May and June, but the sum of the load and operating time values in the hourly records (where MHHIIndicator is not equal to 1) are equal to 0.	Fatal
M	You have reported LTFF as the heat input method for this ozone-season reporting configuration, but you have not reported a long-term fuel flow record for May/June.	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation LME Initialization

Check Code: LME-12

Check Name: Check LTFF Fuel Flow Period Code

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LTFF record:

If ***LME Gen OS*** is equal to true,
    If the Quarter of the reporting period is equal to 2,
        If the FuelFlowPeriodCode is null,
            return result A.

    Otherwise,
        If the FuelFlowPeriodCode is not null,
            return result B.

Otherwise,
    If the FuelFlowPeriodCode is not null,
        return result C.

Results:

Result	Response	Severity
A	You did not report a FuelFlowPeriodCode in the LTFF record for [key]. This value is required for LME units with an ozone-season qualification during the second quarter.	Critical Error Level 1
B	You reported a FuelFlowPeriodCode in the LTFF record for [key], but this value is only appropriate during the second quarter.	Critical Error Level 1
C	You reported a FuelFlowPeriodCode in the LTFF record for [key], but this value is only appropriate for LME units with an ozone-season qualification.	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation LTFF Heat Input Data

Check Code: LME-17

Check Name: Check LTFF Total Heat Input

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LTFF record:

If ***LME Gen LTFF Heat Input*** is not null and is greater than or equal to 0 ,

if ***LME Gen Total Heat Input Array*** for the location is greater than or equal to 0,
add ***LME Gen LTFF Heat Input*** to ***LME Gen LTFF Heat Input Array*** for the location.
If ***LME Gen OS*** is true and the FuelFlowPeriodCode is equal to "A",
add ***LME Gen LTFF Heat Input*** to ***LME Gen LTFF April Heat Input Array*** for the location.

If Location is a common pipe,
If ***LME Gen CP Total Heat Input*** is greater than or equal to 0,
add ***LME Gen LTFF Heat Input*** to ***LME Gen CP Total Heat Input***.
If ***LME Gen OS*** is true and the FuelFlowPeriodCode is equal to "A",
add ***LME Gen LTFF Heat Input*** to ***LME Gen CP April Heat Input***.

Otherwise,
If Location is a common pipe,
set ***LME Gen CP Total Heat Input*** to -1.

Set ***LME Gen LTFF Heat Input Array*** for the location to -1.

Results:

Result	Response	Severity
A	obsolete	No Errors

Usage:

1	Process/Category:	LME Emissions Data Generation LTFF Heat Input Data

Check Code: LME-21

Check Name: Locate Hourly Op Record for LME Unit

Related Former Checks:

Applicability:

Description:

Specifications:

Set ***Current LME Hourly Op Record*** to null.
Set ***Generate LME*** to false.

If ***LME Gen Annual*** = true
    Locate all Monitor Method records for the unit and the hour where the ParameterCode is equal to "SO2M", "NOXM", or "CO2M", and the MethodCode is equal to "LME".
else
    Locate all Monitor Method records for the unit and the hour where the ParameterCode is equal to "NOXM and the MethodCode is equal to "LME".

If found,
    Set ***LME Gen Parameters*** to the list of ParameterCodes in the retrieved records.

Otherwise,
    Set ***LME Gen Parameters*** to null.

Locate an Hourly Op Data record for the unit and the hour.

If found,

    If ***LME Gen Parameters*** is null,
        return result A.

    else

        Set ***Current LME Hourly Op Record*** to the retrieved record.
        Set ***Generate LME*** to true.
        if ***LME Gen Annual*** is equal to false, and the current date is in the month of April,
            return result B.

Otherwise,
    If ***LME Gen Parameters*** is not null, AND
            (***LME Gen Annual*** is equal to true OR the current date is in the months of May thru September),
        return result C.

Results:

Result	Response	Severity
A	There is no active LME method in your monitoring plan.	Critical Error Level 1
B	You reported an LME Hourly record for April, but the unit does not have an annual LME qualification. Emissions for this hour will not be included in the totals reported in the Summary Value record.	Informational Message
C	You did not report an LME Hourly record for the hour.	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation Hourly Emissions Data

Check Code: LME-22

Check Name: Check LME Op Time

Related Former Checks: LME-EXP2

Applicability: LME Check

Description:

Specifications:

If ***Current LME Hourly Op Record*** is not null,

If OpTime is null, or is not between 0 and 1 inclusive,
Set ***LME Gen Total Op Time Array*** for location to -1, ***Generate LME*** to false, and return result A.

Otherwise,
If OpTime is greater than 0, AND ***LME Gen Total Op Time Array*** for location is greater than or equal to 0,
Add 1 to ***LME Gen Total Op Hours Array*** for location.
Add OpTime to ***LME Gen Total Op Time Array*** for location.

If current date in the month of April,
Add 1 to ***LME Gen April Op Hours Array*** for location.
Add OpTime to ***LME Gen April Op Time Array*** for location.

Results:

Result	Response	Severity
A	The [fieldname] reported in the LME Hourly record is missing or invalid.	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation Hourly Emissions Data

Check Code: LME-24

Check Name: Check LME Load Value

Related Former Checks: LME-EXP5

Applicability: LME Check

Description:

Specifications:

If ***Current LME Hourly Op Record*** is not null,

If LoadValue is less than 0,
set ***Generate LME*** to false, and return result A.

else if LoadValue is null,

If OperatingTime is greater than 0,
If ***LME Gen HI Method*** is equal to "LTFF"
set ***Generate LME*** to false, and return result B.

Otherwise
return result C.

else
If OperatingTime is equal to 0,
return result D.

Results:

Result	Response	Severity
A	The [fieldname] reported in the LME Hourly record is invalid.	Critical Error Level 1
B	You did not report a LoadValue in the LME Hourly record.	Critical Error Level 1
C	You did not report a LoadValue in the LME Hourly record.	Non-Critical Error
D	You reported a LoadValue in the LME Hourly record. This field should be blank for a non-operating hour.	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation Hourly Emissions Data

Check Code: LME-26

Check Name: Check LME Load UOM

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

If ***Current LME Hourly Op Record*** is not null,

If LoadValue is null,
If LoadUOMCode is not null,
return result A.

Otherwise,
If LoadUOMCode is not equal to "MW", "KLBHR", or "MMBTUHR",
set ***Generate LME*** to false, and return result A.

Otherwise,
Locate Monitor Load record for the location and hour.

If not found, or more than one record is found, or the MaximumLoadUnitsOfMeasureCode is null,
set ***Generate LME*** to false, and return result B.

Otherwise,
If the LoadUOMCode is not equal to the MaximumLoadUnitsOfMeasureCode in the retrieved record,
set ***Generate LME*** to false, and return result C.

Results:

Result	Response	Severity
A	The [fieldname] reported in the LME Hourly record is missing or invalid.	Critical Error Level 1
B	You have not reported a single, active, valid Monitor Load record in your monitoring plan.	Critical Error Level 1
C	The LoadUOMCode in the LME Hourly record is not consistent with the value in the Monitor Load record in your monitoring plan.	Critical Error Level 1

Usage:

1	Process/Category:	LME Emissions Data Generation Hourly Emissions Data
1	Process/Category:	Emissions Data Entry Screen Evaluation Hourly Op Data Evaluation

Check Code: LME-27

Check Name: Check LME Fuel Code List

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

If ***Current LME Hourly Op Record*** is not null,

If OpTime is greater than 0, and FuelCodeList is null,
set ***Generate LME*** to false, and return result A.

Results:

Result	Response	Severity
A	You did not report a value in the FuelCodeList in the LME Hourly record, but the unit operated during the hour.	Critical Error Level 1

Usage:

1	Process/Category:	LME Emissions Data Generation Hourly Emissions Data
1	Process/Category:	Emissions Data Entry Screen Evaluation Hourly Op Data Evaluation

Check Code: LME-28

Check Name: Calculate Heat Input for LME Unit

Related Former Checks: LME-EXP3B, LME-EXP9A

Applicability: LME Check

Description:

Specifications:

LME Gen Heat Input Record = null
LME Calc Heat Input = null
LME Gen Fuel Code = null

If ***Generate LME*** is equal to true,

If ***Current LME Hourly Op Record***.OperatingTime is greater than 0,

If ***LME Gen HI Method*** is equal to "MHHI" or ***Current LME Hourly Op Record***.MHHIIndicator is equal to 1,

Locate all Monitor Default records for the hour and location where the ParameterCode is equal to "MHHI".

If one record is found, the DefaultValue is greater than 0, and the DefaultUnitsOfMeasureCode is equal to "MMBTUHR".

Calculate ***LME Calc Heat Input*** = DefaultValue * ***Current LME Hourly Op Record***.OpTime, rounded to one decimal place.

Otherwise,

If ***LME Gen Parameters*** contains "SO2M",
set ***LME Gen Total SO2M Array*** for location to -1.
If ***LME Gen Parameters*** contains "NOXM",
set ***LME Gen Total NOXM Array*** for location to -1.
If ***LME Gen Parameters*** contains "CO2M",
set ***LME Gen Total CO2M Array*** for location to -1.
set ***LME Gen Total Heat Input Array*** for location to -1, and return result A.

else if ***LME Gen HI Method*** is equal to "LTFF",

If ***LME Gen CP Total Heat Input*** is greater than or equal to 0, ***LME Gen Total Heat Input Array*** for the location is greater than or equal to 0, AND ***Current LME Hourly Op Record***.HourLoad is greater than or equal to 0,

If ***LME Gen OS*** is equal to true, and the Quarter of the reporting period is equal to 2,

If the current date is in the month of April,

if ***LME Gen April Load*** is greater than 0,

If ***Current LME Hourly Op Record***.HourLoad is equal to 0
Set ***LME Calc Heat Input*** = 0

else

Calculate ***LME Calc Heat Input*** = (***LME Gen CP April Heat Input*** * ***Current LME Hourly Op Record***.HourLoad * ***Current LME Hourly Op Record***.OpTime / ***LME Gen April Load***) + (***LME Gen LTFF April Heat Input Array*** for the location * ***Current LME Hourly Op Record***.HourLoad * ***Current LME Hourly Op Record***.OpTime / ***LME Gen April Load Array*** for the location), and round the result to 1 decimal place.

else if ***LME Gen April Optime*** is greater than 0,

Calculate ***LME Calc Heat Input*** = (***LME Gen CP April Heat Input*** * ***Current LME Hourly Op Record***.OpTime / ***LME Gen April Optime***) + (***LME Gen LTFF April Heat Input Array*** for the location * ***Current LME Hourly Op Record***.OpTime / ***LME Gen LTFF April Op Time Array*** for the location), and round the result to 1 decimal place.

Otherwise,

if ***LME Gen Total Load*** is greater than 0,

If ***Current LME Hourly Op Record***.HourLoad is equal to 0,
Set ***LME Calc Heat Input*** = 0

else

Calculate ***LME Calc Heat Input*** = ((***LME Gen CP Total Heat Input - LME Gen CP April Heat Input***) * ***Current LME Hourly Op Record***.HourLoad * ***Current LME Hourly Op Record***.OpTime / (***LME Gen Total Load - LME Gen April Load***)) + ((***LME Gen LTFF Heat Input Array*** for the location - ***LME Gen LTFF April Heat Input Array*** for the location) * ***Current LME Hourly Op Record***.HourLoad * ***Current LME Hourly Op Record***.OpTime / (***LME Gen Total Load Array*** for the location - ***LME Gen April Load Array*** for the location)), and round the result to 1 decimal place.

else if ***LME Gen Total Optime*** is greater than 0,

Calculate ***LME Calc Heat Input*** = ((***LME Gen CP Total Heat Input - LME Gen CP April Heat Input***) * ***Current LME Hourly Op Record***.OpTime / (***LME Gen Total Optime - LME Gen April Optime***)) + ((***LME Gen LTFF Heat Input Array*** for the location - ***LME Gen LTFF April Heat Input Array*** for the location) * ***Current LME Hourly Op Record***.OpTime / (***LME Gen Total Optime Array*** for the location - ***LME Gen LTFF April Op Time Array*** for the location)), and round the result to 1 decimal place.

Otherwise,

if ***LME Gen Total Load*** is greater than 0,

If ***Current LME Hourly Op Record***.HourLoad is equal to 0,
Set ***LME Calc Heat Input*** = 0

else

Calculate ***LME Calc Heat Input*** = (***LME Gen CP Total Heat Input*** * ***Current LME Hourly Op Record***.HourLoad * ***Current LME Hourly Op Record***.OpTime / ***LME Gen Total Load***) + (***LME Gen LTFF Heat Input Array*** for the location * ***Current LME Hourly Op Record***.HourLoad * ***Current LME Hourly Op Record***.OpTime / ***LME Gen Total Load Array*** for the location), and round the result to 1 decimal place.

else if ***LME Gen Total Optime*** is greater than 0,

Calculate ***LME Calc Heat Input*** = (***LME Gen CP Total Heat Input*** * ***Current LME Hourly Op Record***.OpTime / ***LME Gen Total Optime***) + (***LME Gen LTFF Heat Input Array*** for the location * ***Current LME Hourly Op Record***.OpTime / ***LME Gen LTFF***

Total Op Time Array for the location), and round the result to 1 decimal place.

If ***LME Calc Heat Input*** is not null,

If ***LME Calc Heat Input*** is greater than 999,999.9
If ***LME Gen Parameters*** contains "SO2M",
set ***LME Gen Total SO2M Array*** for location to -1.
If ***LME Gen Parameters*** contains "NOXM",
set ***LME Gen Total NOXM Array*** for location to -1.
If ***LME Gen Parameters*** contains "CO2M",
set ***LME Gen Total CO2M Array*** for location to -1.
set ***LME Calc Heat Input*** to null***, LME Gen Total Heat Input Array*** for location to -1, and return result B.

else if ***LME Calc Heat Input*** is greater than or equal to 0,
LME Gen Heat Input Record.HourID = ***Current LME Hourly Op Record***.HourID
LME Gen Heat Input Record.ParameterCode = "HIT"
LME Gen Heat Input Record.AdjustedHourlyValue = ***LME Calc Heat Input***

If ***Current LME Hourly Op Record***.MHHIIndicator is equal to 1,
LME Gen Heat Input Record.MODCCode = "45"

If ***LME Gen Total Heat Input Array*** for location is greater than or equal to 0,
Add ***LME Calc Heat Input*** to ***LME Gen Total Heat Input Array*** for location.
If current date in the month of April,
Add ***LME Calc Heat Input*** to ***LME Gen April Heat Input Array*** for location.

Otherwise,
If ***LME Gen Parameters*** is not null, AND
(***LME Gen Annual*** is equal to true OR the current date is in the months of May thru September),
If ***LME Gen Parameters*** contains "SO2M",
set ***LME Gen Total SO2M Array*** for location to -1.
If ***LME Gen Parameters*** contains "NOXM",
set ***LME Gen Total NOXM Array*** for location to -1.
If ***LME Gen Parameters*** contains "CO2M",
set ***LME Gen Total CO2M Array*** for location to -1.
set ***LME Gen Total Heat Input Array*** for location to -1.

Results:

Result	Response	Severity
A	You did not report a single, active, valid default record for MHHI in your monitoring plan.	Critical Error Level 1
B	The value calculated for [param] in the LME Hourly record exceeds the maximum value.	Critical Error Level 1

Usage:

1	Process/Category:	LME Emissions Data Generation Hourly Emissions Data

Check Code: LME-29

Check Name: Calculate SO2 Mass for LME Unit

Related Former Checks: LME-EXP3C, LME-EXP9C

Applicability: LME Check

Description:

Specifications:

LME Gen SO2M Record = null

If ***LME Gen Parameters*** contains "SO2M" and ***Current LME Hourly Op Record*** is not null,

If ***Current LME Hourly Op Record***.FuelCodeList is not null,

SO2 Rate = 0
SO2 Fuel = null

For each FuelCode in the ***Current LME Hourly Op Record***.FuelCodeList

Locate all Monitor Default Records for the location and hour where ParameterCode is equal to "SO2R", DefaultPurposeCd = "LM", and FuelCode is equal to the current FuelCode.

If one and only one record is found, the DefaultValue is greater than 0, and DefaultUnitsOfMeasureCode is equal to "LBMMBTU",

If *SO2 Rate* is less than the DefaultValue in the retrieved record,
set *SO2 Rate* to the DefaultValue.
set *SO2 Fuel* to the current FuelCode.

Otherwise,
set ***LME Gen Total SO2M Array*** for location to -1, ***LME Gen Fuel Code*** to the current FuelCode, and return result A.

If ***LME Calc Heat Input*** is not null and is greater than or equal to 0 AND *SO2 Rate* is greater than 0,

Calculate *SO2 Mass* = ***LME Calc Heat Input*** * *SO2 Rate*, and round the result to 1 decimal place.

If *SO2 Mass* is greater than 99,999.9
set ***LME Gen Total SO2M Array*** for location to -1, and return result B.

Otherwise,
LME Gen SO2M Record.HourID = ***Current LME Hourly Op Record***.HourID
LME Gen SO2M Record.ParameterCode = "SO2M"
LME Gen SO2M Record.AdjustedHourlyValue = *SO2 Mass*
LME Gen SO2M Record.FuelCode = *SO2 Fuel*

If ***LME Gen Total SO2M Array*** for location is greater than or equal to 0,
Add *SO2 Mass* to ***LME Gen Total SO2M Array*** for location.

Results:

Result	Response	Severity
A	You did not report a single, active, valid [defparm] default record for FuelCode [fuel] in your monitoring plan.	Critical Error Level 1
B	The value calculated for [param] in the LME Hourly record exceeds the maximum value.	Critical Error Level 1

Usage:

1	Process/Category:	LME Emissions Data Generation Hourly Emissions Data

Check Code: LME-30

Check Name: Calculate NOX Mass for LME Unit

Related Former Checks: LME-EXP3A, LME-EXP9B

Applicability: LME Check

Description:

Specifications:

LME Gen NOXM Record = null

If ***LME Gen Parameters*** contains "NOXM" and ***Current LME Hourly Op Record*** is not null and ***Current LME Hourly Op Record***.OpTime is greater than 0,

If ***Current LME Hourly Op Record***.FuelCodeList is not null,

NOX Rate = 0
NOX Fuel = null

For each FuelCode in the ***Current LME Hourly Op Record***.FuelCodeList

If ***Current LME Hourly Op Record***.OperatingConditionCode is null,

Locate all Monitor Default Records for the location and hour where ParameterCode is equal to "NOXR", DefaultPurposeCd = "LM", OperatingConditionCode = "A", and FuelCode is equal to the current FuelCode.

else if ***Current LME Hourly Op Record***.OperatingConditionCode == "U",

Locate all Monitor Default Records for the location and hour where ParameterCode is equal to "NORX", DefaultPurposeCd = "MD", the OperatingConditionCode is equal to "U", and FuelCode is equal to the current FuelCode.

else

Locate all Monitor Default Records for the location and hour where ParameterCode is equal to "NOXR", DefaultPurposeCd = "LM", the OperatingConditionCode is equal to ***Current LME Hourly Op Record***.OperatingConditionCode, and FuelCode is equal to the current FuelCode.

If one and only one record is found, the DefaultValue is greater than 0, and DefaultUnitsOfMeasureCode is equal to "LBMMBTU",

If *NOX Rate* is less than the DefaultValue in the retrieved record,
set *NOX Rate* to the DefaultValue.
set *NOX Fuel* to the current FuelCode.

Otherwise,
LME Gen Total NOXM Array for location to -1.
Set ***LME Gen Fuel Code*** to the current FuelCode.

If ***Current LME Hourly Op Record***.OperatingConditionCode is null,
return result A.
else
return result B.

If ***LME Calc Heat Input*** is not null and is greater than or equal to 0 AND *NOX Rate* is greater than 0,

Calculate *NOX Mass* = ***LME Calc Heat Input*** * *NOX Rate*, and round the result to 1 decimal place.

If *NOX Mass* is greater than 99,999.9
set ***LME Gen Total NOXM Array*** for location to -1, and return result C.

Otherwise,
LME Gen NOXM Record.HourID = ***Current LME Hourly Op Record***.HourID
LME Gen NOXM Record.ParameterCode = "NOXM"
LME Gen NOXM Record.AdjustedHourlyValue = *NOX Mass*
LME Gen NOXM Record.FuelCode = *NOX Fuel*
LME Gen NOXM Record.OperatingConditionCode = ***Current LME Hourly Op Record***.OperatingConditionCode

If ***LME Gen Total NOXM Array*** for location is greater than or equal to 0,
Add *NOX Mass* to ***LME Gen Total NOXM Array*** for location.
If current date in the month of April,
Add *NOX Mass* to ***LME Gen April NOXM Array*** for location.

Results:

Result	Response	Severity
A	You did not report a single, active, valid [defparm] default record for FuelCode [fuel] in your monitoring plan.	Critical Error Level 1
B	You did not report a single, active, valid [defparm] default record for FuelCode [fuel] OperatingConditionCode [cond] in your monitoring plan.	Critical Error Level 1
C	The value calculated for [param] in the LME Hourly record exceeds the maximum value.	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation Hourly Emissions Data

Check Code: LME-31

Check Name: Calculate CO2 Mass for LME Unit

Related Former Checks: LME-EXP3D, LME-EXP9D

Applicability: LME Check

Description:

Specifications:

LME Gen CO2M Record = null

If ***LME Gen Parameters*** contains "CO2M" and ***Current LME Hourly Op Record*** is not null,

If ***Current LME Hourly Op Record***.FuelCodeList is not null,

CO2 Rate = 0
CO2 Fuel = null

For each FuelCode in the ***Current LME Hourly Op Record***.FuelCodeList

Locate all Monitor Default Records for the location and hour where ParameterCode is equal to "CO2R", DefaultPurposeCd = "LM", and FuelCode is equal to the current FuelCode.

If one and only one record is found, the DefaultValue is greater than 0, and DefaultUnitsOfMeasureCode is equal to "TNMMBTU",

If *CO2 Rate* is less than the DefaultValue in the retrieved record,
set *CO2 Rate* to the DefaultValue.
set *CO2 Fuel* to the current FuelCode.

Otherwise,
set ***LME Gen Total CO2M Array*** for location to -1, ***LME Gen Fuel Code*** to the current FuelCode, and return result A.

If ***LME Calc Heat Input*** is not null and is greater than or equal to 0 AND *CO2 Rate* is greater than 0,

Calculate *CO2 Mass* = ***LME Calc Heat Input*** * *CO2 Rate*, and round the result to 1 decimal place.

If *CO2 Mass* is greater than 99,999,999.9
set ***LME Gen Total CO2M Array*** for location to -1, and return result B.

Otherwise,

LME Gen CO2M Record.HourID = ***Current LME Hourly Op Record***.HourID
LME Gen CO2M Record.ParameterCode = "CO2M"
LME Gen CO2M Record.AdjustedHourlyValue = *CO2 Mass*
LME Gen CO2M Record.FuelCode = *CO2 Fuel*

If ***LME Gen Total CO2M Array*** for location is greater than or equal to 0,
Add *CO2 Mass* to ***LME Gen Total CO2M Array*** for location.

Results:

Result	Response	Severity
A	You did not report a single, active, valid [defparm] default record for FuelCode [fuel] in your monitoring plan.	Critical Error Level 1
B	The value calculated for [param] in the LME Hourly record exceeds the maximum value.	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation Hourly Emissions Data

Check Code: LME-32

Check Name: Calculate HIT Summary Values

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

LME Summary Heat Input Record = null

If location is a common pipe,
tempHIT = ***LME Gen LTFF Heat Input Array*** for the location
tempAprilHIT = ***LME Gen LTFF April Heat Input Array*** for the location

else
tempHIT = If ***LME Gen Total Heat Input Array*** for the location
tempAprilHIT = ***LME Gen April Heat Input Array*** for the location

If tempHIT is greater than or equal to 0,

LME Summary Heat Input Record.MonLocId = current location ID
LME Summary Heat Input Record.ReportingPeriodId = current reporting period ID
LME Summary Heat Input Record.ParameterCode = "HIT"

If ***LME Gen OS*** == true and ***LME Gen Annual*** == false and Quarter of Reporting Period is equal to 2,
LME Summary Heat Input Record.CurrentReportingPeriodTotal = *tempHIT* - *tempAprilHIT*, rounded to 0 decimal places.

else
LME Summary Heat Input Record.CurrentReportingPeriodTotal = *tempHIT*, rounded to 0 decimal places.

If ***LME Gen OS*** == true,
If Quarter of Reporting Period is equal to 2,
LME Summary Heat Input Record.OzoneSeasonToDateTotal = *tempHIT* - *tempAprilHIT*, rounded to 0 decimal places.

else if Quarter of Reporting Period is equal to 3,
LME Summary Heat Input Record.OzoneSeasonToDateTotal = *tempHIT*, rounded to 0 decimal places.

else if Quarter of Reporting Period is equal to 4 AND ***LME Year Start Quarter*** is less than 4,

Locate the ***Op Supp Data*** records for the location WHERE the reporting period is the third quarter of the Year of the Reporting Period and OpTypeCode = "HIT".

If found,
LME Summary Heat Input Record.OzoneSeasonToDateTotal = OpValue in the retrived record.

if Quarter of Reporting Period is equal to 3 or 4 AND ***LME Year Start Quarter*** is less than 3,
Locate the ***Op Supp Data*** records for the location WHERE the reporting period is the second quarter of the Year of the Reporting Period and OpTypeCode = "HITOS".

If found,
add OpValue in the retrieved record to ***LME Summary Heat Input Record***.OzoneSeasonToDateTotal.

If ***LME Gen Annual*** == true,
LME Summary Heat Input Record.YearToDateTotal = ***LME Summary Heat Input***

Record.CurrentReportingPeriodTotal, rounded to 0 decimal places.

If the quarter of the current reporting period is greater than the ***LME Year Start Quarter***,

For each reporting period in the year of the current reporting period and in a quarter that is on or after the ***LME Year Start Quarter*** and is prior to the current reporting period.

Locate the ***Op Supp Data*** records for the location and reporting period WHERE the OpTypeCode = "HIT".

If found,
add OpValue in the retrieved record to ***LME Summary Heat Input Record***.YearToDateTotal.

Results:

Result	Response	Severity
A	The program could not determine ozone-season-to-date totals for [osparam], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation Summary Value Data

Check Code: LME-33

Check Name: Calculate OPTIME Summary Values

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

LME Summary Op Time Record = null

If location is a unit AND ***LME Gen Total Op Time Array*** for the location is greater than or equal to 0,

LME Summary Op Time Record.MonLocId = current location ID
LME Summary Op Time Record.ReportingPeriodId = current reporting period ID
LME Summary Op Time Record.ParameterCode = "OPTIME"

If ***LME Gen OS*** == true and ***LME Gen Annual*** == false and Quarter of Reporting Period is equal to 2,
LME Summary Op Time Record.CurrentReportingPeriodTotal = ***LME Gen Total Op Time Array*** for the location - ***LME Gen April Op Time Array*** for the location.

else
LME Summary Op Time Record.CurrentReportingPeriodTotal = ***LME Gen Total Op Time Array*** for the location.

If ***LME Gen OS*** == true,
If Quarter of Reporting Period is equal to 2,
LME Summary Op Time Record.OzoneSeasonToDateTotal = ***LME Gen Total Op Time Array*** for the location - ***LME Gen April Op Time Array*** for the location.

else if Quarter of Reporting Period is equal to 3,
LME Summary Op Time Record.OzoneSeasonToDateTotal = ***LME Gen Total Op Time Array*** for the location.

else if Quarter of Reporting Period is equal to 4 AND ***LME Year Start Quarter*** is less than 4,

Locate the ***Op Supp Data*** records for the location WHERE the reporting period is the third quarter of the Year of the Reporting Period and OpTypeCode = "OPTIME".

If found,
LME Summary Op Time Record.OzoneSeasonToDateTotal = OpValue in the retrived record.

if Quarter of Reporting Period is equal to 3 or 4 AND ***LME Year Start Quarter*** is less than 3,

Locate the ***Op Supp Data*** records for the location WHERE the reporting period is the second quarter of the current year and OpTypeCode = "OSTIME".

If found,
add OpValue in the retrieved record to ***LME Summary Op Time Record***.OzoneSeasonToDateTotal.

If ***LME Gen Annual*** == true,
LME Summary Op Time Record.YearToDateTotal = ***LME Summary Op Time Record***.CurrentReportingPeriodTotal.

If the quarter of the current reporting period is greater than the ***LME Year Start Quarter***,

For each reporting period in the year of the current reporting period and in a quarter that is on or after the ***LME Year Start Quarter*** and is prior to the current reporting period.

Locate the ***Op Supp Data*** records for the location and reporting period WHERE the OpTypeCode = "OPTIME".

If found,

add OpValue in the retrieved record to ***LME Summary Op Time Record***.YearToDateTotal.

Results:

Result	Response	Severity
A	The program could not determine ozone-season-to-date totals for [osparam], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation Summary Value Data

Check Code: LME-34

Check Name: Calculate OPHOURS Summary Values

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

LME Summary Op Hours Record = null

If location is a unit AND ***LME Gen Total Op Time Array*** for the location is greater than or equal to 0,

LME Summary Op Hours Record.MonLocId = current location ID
LME Summary Op Hours Record.ReportingPeriodId = current reporting period ID
LME Summary Op Hours Record.ParameterCode = "OPHOURS"

If ***LME Gen OS*** == true and ***LME Gen Annual*** == false and Quarter of Reporting Period is equal to 2,
LME Summary Op Hours Record.CurrentReportingPeriodTotal = ***LME Gen Total Op Hours Array*** for the location - ***LME Gen April Op Hours Array*** for the location.
else
LME Summary Op Hours Record.CurrentReportingPeriodTotal = ***LME Gen Total Op Hours Array*** for the location.

If ***LME Gen OS*** == true,
If Quarter of Reporting Period is equal to 2,
LME Summary Op Hours Record.OzoneSeasonToDateTotal = ***LME Gen Total Op Hours Array*** for the location - ***LME Gen April Op Hours Array*** for the location.

else if Quarter of Reporting Period is equal to 3,
LME Summary Op Hours Record.OzoneSeasonToDateTotal = ***LME Gen Total Op Hours Array*** for the location.

else if Quarter of Reporting Period is equal to 4 AND ***LME Year Start Quarter*** is less than 4,

Locate the ***Op Supp Data*** records for the location WHERE the reporting period is the third quarter of the Year of the Reporting Period and OpTypeCode = "OPHOURS".

If found,
LME Summary Op Hours Record.OzoneSeasonToDateTotal = OpValue in the retrived record.

if Quarter of Reporting Period is equal to 3 or 4 AND ***LME Year Start Quarter*** is less than 3,

Locate the ***Op Supp Data*** records for the location WHERE the reporting period is the second quarter of the current year and OpTypeCode = "OSHOURS".

If found,
add OpValue in the retrieved record to ***LME Summary Op Hours Record***.OzoneSeasonToDateTotal.

If ***LME Gen Annual*** == true,
LME Summary Op Hours Record.YearToDateTotal = ***LME Summary Op Hours Record***.CurrentReportingPeriodTotal.

If the quarter of the current reporting period is greater than the ***LME Year Start Quarter***,

For each reporting period in the year of the current reporting period and in a quarter that is on or after the ***LME Year Start Quarter*** and is prior to the current reporting period.

Locate the ***Op Supp Data*** records for the location and reporting period WHERE the OpTypeCode = "OPHOURS".

If found,

add OpValue in the retrieved record to ***LME Summary Op Hours Record***.YearToDateTotal.

Results:

Result	Response	Severity
A	The program could not determine ozone-season-to-date totals for [osparam], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1

Usage:

1	Process/Category:	LME Emissions Data Generation Summary Value Data

Check Code: LME-35

Check Name: Calculate SO2M Summary Values

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

LME Summary SO2M Record = null

If location is a unit, ***LME Gen Annual*** == true, AND ***LME Gen Total SO2M Array*** for the location is greater than or equal to 0,

Locate a Monitor Method record for the unit where the ParameterCode is equal to "SO2M", and the MethodCode is equal to "LME", BeginDate is on or before the first day of the reporting period, and the EndDate is null or is on or after the last day of the reporting period.

If found,

LME Summary SO2M Record.MonLocId = current location ID
LME Summary SO2M Record.ReportingPeriodId = current reporting period ID
LME Summary SO2M Record.ParameterCode = "SO2M"
LME Summary SO2M Record.CurrentReportingPeriodTotal = ***LME Gen Total SO2M Array*** for the location / 2000, rounded to one decimal place.
LME Summary SO2M Record.YearToDateTotal = ***LME Summary SO2M Record***.CurrentReportingPeriodTotal.

If the quarter of the current reporting period is greater than the ***LME Year Start Quarter***,

For each reporting period in the year of the current reporting period and in a quarter that is on or after the ***LME Year Start Quarter*** and is prior to the current reporting period.

Locate the ***Op Supp Data*** records for the location and reporting period WHERE the OpTypeCode = "SO2M".

If found,
add OpValue in the retrieved record to ***LME Summary SO2M Record***.YearToDateTotal.

Results:

Result	Response	Severity
A	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation Summary Value Data

Check Code: LME-36

Check Name: Calculate CO2M Summary Values

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

LME Summary CO2M Record = null

If location is a unit, ***LME Gen Annual*** == true, AND ***LME Gen Total CO2M Array*** for the location is greater than or equal to 0,

Locate a Monitor Method record for the unit where the ParameterCode is equal to "CO2M", and the MethodCode is equal to "LME", BeginDate is on or before the first day of the reporting period, and the EndDate is null or is on or after the last day of the reporting period.

If found,

LME Summary CO2M Record.MonLocId = current location ID
LME Summary CO2M Record.ReportingPeriodId = current reporting period ID
LME Summary CO2M Record.ParameterCode = "CO2M"
LME Summary CO2M Record.CurrentReportingPeriodTotal = ***LME Gen Total CO2M Array*** for the location.
LME Summary CO2M Record.YearToDateTotal = ***LME Summary CO2M Record***.CurrentReportingPeriodTotal.

If the quarter of the current reporting period is greater than the ***LME Year Start Quarter***,

For each reporting period in the year of the current reporting period and in a quarter that is on or after the ***LME Year Start Quarter*** and is prior to the current reporting period.

Locate the ***Op Supp Data*** records for the location and reporting period WHERE the OpTypeCode = "CO2M".

If found,
add OpValue in the retrieved record to ***LME Summary CO2M Record***.YearToDateTotal.

Results:

Result	Response	Severity
A	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation Summary Value Data

Check Code: LME-37

Check Name: Calculate NOXM Summary Values

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

LME Summary NOXM Record = null

If location is a unit AND ***LME Gen Total NOXM Array*** for the location is greater than or equal to 0,

Locate a Monitor Method record for the unit where:

1) ParameterCode is equal to "NOXM".
2) MethodCode is equal to "LME".
3) BeginDate is on or before:
a) May 1st of the year of the reporting period when the reporting period is for the 2nd quarter AND ***LME Gen OS*** is equal to true.
b) The first day of the reporting period otherwise.
4) EndDate is null or is on or after the last day of the reporting period.

If found,

LME Summary NOXM Record.MonLocId = current location ID
LME Summary NOXM Record.ReportingPeriodId = current reporting period ID
LME Summary NOXM Record.ParameterCode = "NOXM"

If ***LME Gen OS*** == true and ***LME Gen Annual*** == false and Quarter of Reporting Period is equal to 2,
LME Summary NOXM Record.CurrentReportingPeriodTotal = (***LME Gen Total NOXM Array*** for the location - ***LME Gen April NOXM Array*** for the location) / 2000, rounded to one decimal place.
else
LME Summary NOXM Record.CurrentReportingPeriodTotal = ***LME Gen Total NOXM Array*** for the location / 2000, rounded to one decimal place.

If ***LME Gen OS*** == true,
If Quarter of Reporting Period is equal to 2,
LME Summary NOXM Record.OzoneSeasonToDateTotal = (***LME Gen Total NOXM Array*** for the location - ***LME Gen April NOXM Array*** for the location) / 2000, rounded to one decimal place.

else if Quarter of Reporting Period is equal to 3,
LME Summary NOXM Record.OzoneSeasonToDateTotal = ***LME Gen Total NOXM Array*** for the location / 2000, rounded to one decimal place.

else if Quarter of Reporting Period is equal to 4 AND ***LME Year Start Quarter*** is less than 4,

Locate the ***Op Supp Data*** records for the location WHERE the reporting period is the third quarter of the Year of the Reporting Period and OpTypeCode = "NOXM".

If found,
LME Summary NOXM Record.OzoneSeasonToDateTotal = OpValue in the retrived record.

if Quarter of Reporting Period is equal to 3 or 4 AND ***LME Year Start Quarter*** is less than 3,
Locate the ***Op Supp Data*** records for the location WHERE the reporting period is the second quarter of the current year and OpTypeCode = "NOXMOS".

If found,

add OpValue in the retrieved record to ***LME Summary NOXM Record***.OzoneSeasonToDateTotal.

If ***LME Gen Annual*** == true,
LME Summary NOXM Record.YearToDateTotal = ***LME Summary NOXM Record***.CurrentReportingPeriodTotal.

If the quarter of the current reporting period is greater than the ***LME Year Start Quarter***,

For each reporting period in the year of the current reporting period and in a quarter that is on or after the ***LME Year Start Quarter*** and is prior to the current reporting period.

Locate the ***Op Supp Data*** records for the location and reporting period WHERE the OpTypeCode = "NOXM".

If found,
add OpValue in the retrieved record to ***LME Summary NOXM Record***.YearToDateTotal.

Results:

Result	Response	Severity
A	The program could not determine ozone-season-to-date totals for [osparam], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1

Usage:

1	Process/Category:	LME Emissions Data Generation Summary Value Data

Check Code: LME-38

Check Name: Calculate NOXR Summary Values

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

LME Summary NOXR Record = null

If ***LME Summary Heat Input Record*** and ***LME Summary NOXM Record*** are both not null,

Locate a Program record for the unit where the ProgramCode is equal to "ARP", the Class is not equal to "NA", UnitMonitorCertBeginDate is on or before the last day of the reporting period, and the EndDate is null or is on or after the first day of the reporting period.

If found,

LME Summary NOXR Record.MonLocId = current location ID
LME Summary NOXR Record.ReportingPeriodId = current reporting period ID
LME Summary NOXR Record.ParameterCode = "NOXR"

If ***LME Summary NOXM Record***.ReportingPeriodTotal == 0
LME Summary NOXR Record.CurrentReportingPeriodTotal = 0
else
LME Summary NOXR Record.CurrentReportingPeriodTotal = ***LME Gen Total NOXM Array*** for the location / ***LME Summary Heat Input Record***.ReportingPeriodTotal, rounded to 3 decimal places.

If the quarter of the current reporting period is greater than the ***LME Year Start Quarter***,

If ***LME Summary Heat Input Record***.YearToDateTotal is not null,

NOxTotal = ***LME Gen Total NOXM Array*** for the location.

For each reporting period in the year of the current reporting period and in a quarter that is on or after the ***LME Year Start Quarter*** and is prior to the current reporting period.

Locate the ***Op Supp Data*** records for the location and reporting period WHERE the OpTypeCode = "NOXR".

If found,
NOXRValue = OpValue.

Locate the ***Op Supp Data*** records for the location and reporting period WHERE the OpTypeCode = "HIT".

If found,
Add OpValue * *NOXRValue* (rounded to 1 decimal) to *NOxTotal.*

If *NOxTotal* == 0,
LME Summary NOXR Record.YearToDateTotal = 0.

else
LME Summary NOXR Record.YearToDateTotal = *NOxTotal* / ***LME Summary Heat Input Record***.YearToDateTotal, rounded to 3 decimal places.

else

LME Summary NOXR Record.YearToDateTotal = ***LME Summary NOXR Record***.CurrentReportingPeriodTotal.

Results:

Result	Response	Severity
A	The program could not determine year-to-date for [param], because the Op Supp Data record for this parameter is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
B	The program could not determine year-to-date for [param], because the Op Supp Data record for HIT is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1

Usage:

1	Process/Category:	LME Emissions Data Generation Summary Value Data

Check Code: LME-40

Check Name: Check LME MHHI Indicator

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

If ***Current LME Hourly Op Record*** is not null,

If MHHIIndicator is equal to 1,
If ***LME Gen HI Substitute Data*** is not equal to "MHHI",
set ***Generate LME*** to false, and return result A.

Results:

Result	Response	Severity
A	You reported a MHHIIndicator, but you did not report an active LTFF heat input method with a SubstituteDataCode of "MHHI".	Critical Error Level 1

Usage:

1 Process/Category: LME Emissions Data Generation Hourly Emissions Data

Check Code: LME-44

Check Name: Check Fuel Codes against LTFF Records

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

If ***Current LME Hourly Op Record*** is not null and ***LME Gen HI Method*** = "LTFF",

If MHHIIndicator is not equal to 1 and FuelCodeList is not null,

Locate all ***Unit Stack Configuration Records*** where the unit location is the location in the ***Current LME Hourly Op Record***, the StackID begins with "CP", the BeginDate and BeginHour is on or before the Date and Hour in the ***Current LME Hourly Op Record***, and the EndDate is null or the EndDate and EndHour is on or after the Date and Hour in the ***Current LME Hourly Op Record***.

For each FuelCode in FuelCodeList,

Locate a ***LTFF Record*** for the configuration and reporting period where the location is the location in the ***Current LME Hourly Op Record*** or is any of the common pipes in the retrieved ***Unit Stack Configuration Records,*** and the FuelCode of the associated fuel flow system is equal to the FuelCode in the FuelCodeList that is being evaluated.

If not found for any FuelCode,
set ***Generate LME*** to false, and return result A.

Results:

Result	Response	Severity
A	You did not report a Long Term Fuel Flow record for a fuel flow system for one or more fuels in the FuelCodeList in the LME Hourly record. If you burn a fuel that is not measured by a fuel flow meter, you must use maximum hourly heat input for the hour.	Critical Error Level 1

Usage:

1	Process/Category:	LME Emissions Data Generation Hourly Emissions Data

Check Code: LME-45

Check Name: Duplicate LTFF Record

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LTFF record:

Locate another LTFF record for the location with same ReportingPeriod, MonitoringSystemID, and FuelFlowPeriodCode as the current record.

If found,
return result A.

Results:

Result	Response	Severity
A	Another [recordtype] record already exists with the same [fieldnames].	Fatal

Usage:

1	Process/Category:	LME Emissions Data Generation LTFF Heat Input Data

Check Code: LME-18

Check Name: Check LME Begin Hour

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LME Hourly Op record:

If BeginHour is null or is not between 0 and 23 (inclusive),
return result A.

Results:

Result	Response	Severity
A	The Hour is missing or invalid.	Fatal

Usage:

1 Process/Category: Emissions Data Entry Screen Evaluation Hourly Op Data Evaluation

Check Code: LME-19

Check Name: Check LME Begin Date

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LME Hourly Op record:

If BeginDate is null or is not within the reporting period,
return result A.

Results:

Result	Response	Severity
A	The Date is missing or not within the reporting period.	Fatal

Usage:

1	Process/Category:	Emissions Data Entry Screen Evaluation Hourly Op Data Evaluation

Check Code: LME-20

Check Name: Duplicate LME Hourly Op Record

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LME Hourly Op record:

Locate another Hourly Op record for the location with same BeginDate and BeginHour.

If found,
return result A.

Results:

Result	Response	Severity
A	Another [recordtype] record already exists with the same [fieldnames].	Fatal

Usage:

1 Process/Category: Emissions Data Entry Screen Evaluation Hourly Op Data Evaluation

Check Code: LME-23

Check Name: Check LME Data Entry Screen Op Time

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LME Hourly Op record:

If OpTime is null, or is not between 0 and 1 inclusive,
return result A.

Results:

Result	Response	Severity
A	The [fieldname] reported in the LME Hourly record is missing or invalid.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Entry Screen Evaluation Hourly Op Data Evaluation

Check Code: LME-25

Check Name: Check LME Data Entry Screen Load Value

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LME Hourly Op record:

If LoadValue is null,
    If OperatingTime is greater than 0,
        return result A.

else if LoadValue is less than 0,
    return result A.

else
    If OperatingTime is equal to 0,
        return result B.

Results:

Result	Response	Severity
A	The [fieldname] reported in the LME Hourly record is missing or invalid.	Critical Error Level 1
B	You reported a LoadValue in the LME Hourly record. This field should be blank for a non-operating hour.	Non-Critical Error

Usage:

1 Process/Category: Emissions Data Entry Screen Evaluation Hourly Op Data Evaluation

Check Code: LME-43

Check Name: Check LTFF Fuel Flow Period Code

Related Former Checks:

Applicability: LME Check

Description:

Specifications:

For the LTFF record:

If the Quarter of the reporting period is not equal to 2,
If the FuelFlowPeriodCode is not null,
return result A.

Results:

Result	Response	Severity
A	You reported a FuelFlowPeriodCode in the LTFF record for [key], but this value is only appropriate during the second quarter.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Entry Screen Evaluation LTFF Data Evaluation

Check Category:

MATS Calculated Hourly Value Checks

Check Code: MATSCHV-1

Check Name: Initialize HGRE Calculated Hourly Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for HGRE.

Specifications:

CalculationConversionFactor = 6.24 x 10 ^ -11
CurrentDhvParameter = ***MatsHgDhvParameter***
CurrentDhvRecordValid = ***MatsHgDhvValid***
MatsDhvRecord = MatsHgDhvRecord
MatsMhvCalculatedValue = ***MatsMhvCalculatedHgcValue***
MatsMhvRecord = MatsHgcMhvRecord
MatsMoistureEquationList = {A-3}

MatsDhvMeasuredModcList to {36, 39}
MatsDhvUnavailableModcList to {38}

If ***CurrentHourlyOpRecord***.MatsHourLoad is NOT equal to 0, AND is NOT null,
    FinalConversionFactor = 10^3 / ***CurrentHourlyOpRecord***.MatsHourLoad
Else
    FinalConversionFactor = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Calculation Verification

Check Code: MATSCHV-2

Check Name: Initialize HCLRE Calculated Hourly Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for HCLRE.

Specifications:

CalculationConversionFactor = 9.43 x 10 ^ -8
CurrentDhvParameter = ***MatsHclDhvParameter***
CurrentDhvRecordValid = ***MatsHclDhvValid***
MatsDhvRecord = MatsHclDhvRecord
MatsMhvCalculatedValue = ***MatsMhvCalculatedHclcValue***
MatsMhvRecord = MatsHclcMhvRecord
MatsMoistureEquationList = {HC-3}

MatsDhvMeasuredModcList to {36, 39}
MatsDhvUnavailableModcList to {38}

If ***CurrentHourlyOpRecord***.MatsHourLoad is NOT equal to 0, AND is NOT null,
    FinalConversionFactor = 1 / ***CurrentHourlyOpRecord***.MatsHourLoad
Else
    FinalConversionFactor = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Calculation Verification

Check Code: MATSCHV-3

Check Name: Initialize HFRE Calculated Hourly Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for HFRE.

Specifications:

CalculationConversionFactor = 5.18 x 10 ^ -8
CurrentDhvParameter = ***MatsHfDhvParameter***
CurrentDhvRecordValid = ***MatsHfDhvValid***
MatsDhvRecord = MatsHfDhvRecord
MatsMhvCalculatedValue = ***MatsMhvCalculatedHfcValue***
MatsMhvRecord = MatsHfcMhvRecord
MatsMoistureEquationList = {HF-3}

MatsDhvMeasuredModcList to {36, 39}
MatsDhvUnavailableModcList to {38}

If ***CurrentHourlyOpRecord***.MatsHourLoad is NOT equal to 0, AND is NOT null,
    FinalConversionFactor = 1 / ***CurrentHourlyOpRecord***.MatsHourLoad
Else
    FinalConversionFactor = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Calculation Verification

Check Code: MATSCHV-4

Check Name: Initialize SO2RE Calculated Hourly Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for SO2RE.

Specifications:

CalculationConversionFactor = 1.66 x 10 ^ -7
CurrentDhvParameter = ***MatsSo2DhvParameter***
CurrentDhvRecordValid = ***MatsSo2DhvValid***
MatsDhvRecord = MatsSo2DhvRecord
MatsMoistureEquationList = {S-3}

MatsDhvMeasuredModcList to {36, 39}
MatsDhvUnavailableModcList to {38}

If ***CurrentHourlyOpRecord***.MatsHourLoad is NOT equal to 0, AND is NOT null,
    FinalConversionFactor = 1 / ***CurrentHourlyOpRecord***.MatsHourLoad
Else
    FinalConversionFactor = null

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Calculation Verification

Check Code: MATSCHV-5

Check Name: Initialize HGRH Calculated Hourly Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for HGRH.

Specifications:

CalculationConversionFactor = 6.24 x 10 ^ -11
CurrentDhvParameter = ***MatsHgDhvParameter***
CurrentDhvRecordValid = ***MatsHgDhvValid***
MatsDhvRecord = MatsHgDhvRecord
MatsMhvCalculatedValue = ***MatsMhvCalculatedHgcValue***
MatsMhvRecord = MatsHgcMhvRecord
MatsMoistureEquationList = {19-3, 19-3D, 19-4, 19-5, 19-8, 19-9}

MatsDhvMeasuredModcList to {36, 37}
MatsDhvUnavailableModcList to {38}

FinalConversionFactor = 10^6

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Calculation Verification

Check Code: MATSCHV-6

Check Name: Initialize HCLRH Calculated Hourly Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for HCLRH.

Specifications:

CalculationConversionFactor = 9.43 x 10 ^ -8
CurrentDhvParameter = ***MatsHclDhvParameter***
CurrentDhvRecordValid = ***MatsHclDhvValid***
MatsDhvRecord = MatsHclDhvRecord
MatsMhvCalculatedValue = ***MatsMhvCalculatedHclcValue***
MatsMhvRecord = MatsHclcMhvRecord
MatsMoistureEquationList = {19-3, 19-3D, 19-4, 19-5, 19-8, 19-9}

MatsDhvMeasuredModcList to {36, 37}
MatsDhvUnavailableModcList to {38}

FinalConversionFactor = 1

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Calculation Verification

Check Code: MATSCHV-7

Check Name: Initialize HFRH Calculated Hourly Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for HFRH.

Specifications:

CalculationConversionFactor = 5.18 x 10 ^ -8
CurrentDhvParameter = ***MatsHfDhvParameter***
CurrentDhvRecordValid = ***MatsHfDhvValid***
MatsDhvRecord = MatsHfDhvRecord
MatsMhvCalculatedValue = ***MatsMhvCalculatedHfcValue***
MatsMhvRecord = MatsHfMhvRecord
MatsMoistureEquationList = {19-3, 19-3D, 19-4, 19-5, 19-8, 19-9}

MatsDhvMeasuredModcList to {36, 37}
MatsDhvUnavailableModcList to {38}

FinalConversionFactor = 1

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Calculation Verification

Check Code: MATSCHV-8

Check Name: Initialize SO2RH Calculated Hourly Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent Calculated hourly checks for SO2RH.

Specifications:

CalculationConversionFactor = 1.66 x 10 ^ -7
CurrentDhvParameter = ***MatsSo2DhvParameter***
CurrentDhvRecordValid = ***MatsSo2DhvValid***
MatsDhvRecord = MatsSo2DhvRecord
MatsMoistureEquationList = {19-3, 19-3D, 19-4, 19-5, 19-8, 19-9}

MatsDhvMeasuredModcList to {36, 37}
MatsDhvUnavailableModcList to {38}

FinalConversionFactor = 1

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Calculation Verification

Check Code: MATSCHV-9

Check Name: Determine the Calculation Concentration for a MATS Parameter

Related Former Checks:

Applicability:

Description: Determines the main concentration value to use in calculations.

Specifications:

CalculationConcentration = null
CalculationConcentrationSubstituted = false

If ***CurrentDhvRecordValid*** AND (***MatsDhvRecord***.ModcCode in ***MatsDhvMeasuredModcList***)

If (***MatsMhvCalculatedValue*** is not null)
CalculationConcentration = MatsMhvCalculatedValue (convert from Scientific Notation)

If (***MatsMhvRecordMatsMhvRecord*** is not null) AND (***MatsMhvRecord***.ModcCode is equal to "34" or "35")
CalculationConcentrationSubstituted = true

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Calculation Verification
3	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Calculation Verification
4	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Calculation Verification
5	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Calculation Verification
6	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Calculation Verification

Check Code: MATSCHV-10

Check Name: Determine the Calculation Concentration for SO2

Related Former Checks:

Applicability:

Description: Determines the SO2 concentration value to use in calculations.

Specifications:

CalculationConcentration = null
CalculationConcentrationSubstituted = false

If ***CurrentDhvRecordValid*** AND (***MatsDhvRecord***.ModcCode in ***MatsDhvMeasuredModcList***)

If ***CurrentSo2MonitorHourlyRecord*** is not null

CalculationConcentration = CurrentSo2MonitorHourlyRecord.UnadjustedHourlyValue

If (***CurrentSo2MonitorHourlyRecord***.ModcCode in set {05, 06, 07, 08, 09, 10, 12, 13, 15, 18, 23, 55})
CalculationConcentrationSubstituted = true

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Calculation Verification

Check Code: MATSCHV-11

Check Name: Determine the Calculation Flow

Related Former Checks:

Applicability:

Description: Determines the flow value for equations.

Specifications:

CalculationFlow = null
CalculationFlowSubstituted = false

If ***CurrentDhvRecordValid*** AND (***MatsDhvRecord***.ModcCode in ***MatsDhvMeasuredModcList***)

If (***CurrentStackFlowHourlyRecord*** is NOT null)

CalculationFlow = ***CurrentStackFlowHourlyRecord***.UnadjustedHourlyValue

If (***CurrentStackFlowHourlyRecord***.ModcCode not in set {01, 02, 03, 04, 20, 53, 54})
CalculationFlowSubstituted = true

Else

CalculationFlow = null
CalculationFlowSubstituted = false

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Calculation Verification
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Calculation Verification
4	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Calculation Verification

Check Code: MATSCHV-12

Check Name: Determine the Calculation Diluent Value

Related Former Checks:

Applicability:

Description: Determines the diluent value to use in calculations involving formula 19 equations.

Specifications:

CalculationDiluent = null
CalculationDiluentSubstituted = false

If ***CurrentDhvRecordValid*** AND (***MatsDhvRecord***.ModcCode in ***MatsDhvMeasuredModcList***)

If (***MatsDhvRecord***.EquationCode in set {19-3D, 19-5D} OR ***MatsDhvRecord***.ModcCode == 37)

If (***MatsDhvRecord***.EquationCode in set {19-1, 19-2, 19-3, 19-3D, 19-4, 19-5, 19-5D})

O2MonitorDefaultMatches = count of ***MonitorDefaultRecordsByHourLocation*** where:

1) ParameterCode = "O2X"
2) DefaultPurposeCode = "DC"
3) FuelCode = "NFS"

If (*O2MonitorDefaultMatches* > 1)
return result A

Else if (*O2MonitorDefaultMatches* == 0)
return result B

Else
O2MonitorDefaultRecord = the single matched record

If (*O2MonitorDefaultRecord*.DefaultValue is NULL OR *O2MonitorDefaultRecord*.DefaultValue <= 0)
return result C
Else
CalculationDiluent = *O2MonitorDefaultRecord*.DefaultValue

Else if (***MatsDhvRecord***.EquationCode in set {19-6, 19-7, 19-8, 19-9})

Co2MonitorDefaultMatches = count of ***MonitorDefaultRecordsByHourLocation*** where:

1) ParameterCode = "CO2N"
2) DefaultPurposeCode = "DC"
3) FuelCode = "NFS"

If (*Co2MonitorDefaultMatches* > 1)
return result D

Else if (*Co2MonitorDefaultMatches* == 0)
return result E

Else
CO2MonitorDefaultRecord = the single matched record

If *(Co2MonitorDefaultRecord*.DefaultValue is NULL OR *Co2MonitorDefaultRecord*.DefaultValue <= 0)
return result F
Else

CalculationDiluent=*Co2MonitorDefaultRecord*.DefaultValue

Else

If (***MatsDhvRecord***.EquationCode in set { 19-1, 19-4} AND ***O2DryNeededForMats*** == true)
CalculationDiluent = ***O2DryCalculatedAdjustedValue***

If (***O2DryModc*** not in set {01, 02, 03, 04, 17, 20, 53, 54})
CalculationDiluentSubstituted = true

Else if (***MatsDhvRecord***.EquationCode in set {19-2, 19-3, 19-5} AND ***O2WetNeededForMats*** == true)
CalculationDiluent = ***O2WetCalculatedAdjustedValue***

If (***O2WetModc*** not in set {01, 02, 03, 04, 17, 20, 53, 54})
CalculationDiluentSubstituted = true

Else if (***MatsDhvRecord***.EquationCode in set { 19-6, 19-7, 19-8, 19-9} AND ***Co2DiluentNeededForMats*** == true)
CalculationDiluent = ***Co2cMhvCalculatedAdjustedValue***

If (***Co2cMhvModc*** not in set {01, 02, 03, 04, 17, 20, 21, 53, 54})
CalculationDiluentSubstituted = true

Results:

Result	Response	Severity
A	You reported more than one diluent cap default record for O2X in your monitoring plan that was active during current hour.	Critical Error Level 1
B	You did not report a default record for O2X in your monitoring plan that was active during current hour.	Critical Error Level 1
C	The DefaultValue reported in the active Default record for O2X in your monitoring plan is invalid. The value must be greater than 0.	Critical Error Level 1
D	You reported more than one diluent cap default record for CO2N in your monitoring plan that was active during the current hour.	Critical Error Level 1
E	You did not report an active CO2N diluent cap default record in your monitoring plan for the hour.	Critical Error Level 1
F	The DefaultValue reported in the active Default record for CO2N in your monitoring plan is invalid. The value must be greater than 0.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Calculation Verification
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Calculation Verification
4	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Calculation Verification

Check Code: MATSCHV-13

Check Name: Determine the Calculation Moisture

Related Former Checks:

Applicability:

Description: Determines the moisture value for equations.

Specifications:

CalculationMoisture = null
CalculationMoistureSubstituted = false

If ***CurrentDhvRecordValid*** AND (***MatsDhvRecord***.ModcCode in ***MatsDhvMeasuredModcList***)

If (***MatsDhvRecord***.EquationCode in ***MatsMoistureEquationList***)

If (***H2oMethodCode*** is equal to "MWD") AND ***H2oDerivedHourlyChecksNeeded*** AND (***H2oDhvCalculatedAdjustedValue*** is not null)
CalculationMoisture = H2oDhvCalculatedAdjustedValue

If (***H2oDhvModc*** not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 12, 21, 53, 54, 55})
CalculationMoistureSubstituted = true

Else if (***H2oMethodCode*** in set {MMS, MTB}) AND ***H2oMonitorHourlyChecksNeeded*** AND (***H2oMhvCalculatedAdjustedValue*** is not null)
CalculationMoisture = H2oMhvCalculatedAdjustedValue

If (***H2oMhvModc*** not in set {01, 02, 03, 04, 06, 07, 08, 09, 10, 12, 21, 53, 54, 55})
CalculationMoistureSubstituted = true

Else if (***H2oMethodCode*** is equal to "MDF") AND ***H2oDerivedHourlyChecksNeeded*** AND (***H2oDhvCalculatedAdjustedValue*** is not null)
CalculationMoisture = H2oDhvCalculatedAdjustedValue

If (***H2oDhvModc*** not in set {01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 12, 21, 53, 54, 55})
CalculationMoistureSubstituted = true

Else if (***H2oMethodCode*** is equal to "MDF") AND (***H2oDerivedHourlyChecksNeeded*** is false) AND (***H2oDefaultValue*** is not null)
CalculationMoisture = H2oDefaultValue

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Calculation Verification
3	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Calculation Verification
4	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Calculation Verification
5	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Calculation Verification
6	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Calculation Verification
7	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Calculation Verification
8	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Calculation Verification

Check Code: MATSCHV-14

Check Name: Check MODC and determine the MATS Formula Calculated Unadjusted Value

Related Former Checks:

Applicability:

Description: Calculates the Unadjusted Value using MATS formula equations.

Specifications:

CalculatedUnadjustedValue = null

If ***CurrentDhvRecordValid***

If (***MatsDhvRecord.***ModcCode in ***MatsDhvMeasuredModcList***)

If (***MatsDhvRecord***.EquationCode is NOT null)

If (***MatsDhvRecord***.EquationCode is in ***MatsMoistureEquationList***)

If ***CalculationConcentrationSubstituted*** OR ***CalculationFlowSubstituted*** OR ***CalculationMoistureSubstituted***

return result A

Else if (***CalculationConcentration*** is null) OR (***CalculationFlow*** is null) OR (***CalculationMoisture*** is null)

return result B

Else if (***FinalConversionFactor*** is NOT null)
CalculatedUnadjustedValue = (***CalculationConversionFactor*** * ***CalculationConcentration*** * ***CalculationFlow*** * (1 - ***CalculationMoisture*** / 100)) * ***FinalConversionFactor***

Else

If ***CalculationConcentrationSubstituted*** OR ***CalculationFlowSubstituted***

return result C

Else if (***CalculationConcentration*** is null) OR (***CalculationFlow*** is null)

return result D

Else if (***FinalConversionFactor*** is NOT null)
CalculatedUnadjustedValue = (***CalculationConversionFactor*** * ***CalculationConcentration*** * ***CalculationFlow***) * ***FinalConversionFactor***

Else

return result E

Results:

Result	Response	Severity
A	You reported an incorrect MODCCode in the MATS Derived Hourly Value record for [param], as you reported substitute data for one or more essential auxiliary parameters.	Critical Error Level 1
B	You reported a FormulaIdentifier in the MATS Derived Hourly Value record for [param], but you did not report a value for all the essential parameters needed to perform the calculation.	Critical Error Level 1
C	You reported an incorrect MODCCode in the MATS Derived Hourly Value record for [param], as you reported substitute data for one or more essential auxiliary parameters.	Critical Error Level 1
D	You reported a FormulaIdentifier in the MATS Derived Hourly Value record for [param], but you did not report a value for all the essential parameters needed to perform the calculation.	Critical Error Level 1
E	You reported an incorrect MODCCode in the MATS Derived Hourly Value record for [param], as you reported measured data for essential auxiliary parameters.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Calculation Verification
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Calculation Verification
4	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Calculation Verification

Check Code: MATSCHV-15

Check Name: Check MODC and determine the Formula 19 Calculated Unadjusted Value

Related Former Checks:

Applicability:

Description: Calculates the Unadjusted Value using Formula 19 equations.

Specifications:

CalculatedUnadjustedValue = null

If ***CurrentDhvRecordValid***

If (***MatsDhvRecord.***ModcCode in ***MatsDhvMeasuredModcList***)

If (***MatsDhvRecord***.EquationCode is NOT null)

Case (***MatsDhvRecord***.EquationCode)

"19-1" :

If ***CalculationConcentrationSubstituted*** OR ***CalculationDiluentSubstituted***

return result A

Else if (***CalculationDiluent*** is null) OR (***CalculationConcentration*** is null) OR (***ValidFdFactorExists*** is false)

return result C

Else if (***CalculationDiluent*** is equal to 20.9)

return result D

Else if (***FinalConversionFactor*** is NOT null)

CalculatedUnadjustedValue = (***CalculationConversionFactor*** * ***CalculationConcentration*** * ***CurrentHourlyOpRecord***.FdFactor * [20.9 / (20.9 - ***CalculationDiluent***)]) * ***FinalConversionFactor***

"19-2":

If ***CalculationConcentrationSubstituted*** OR ***CalculationDiluentSubstituted***

return result A

Else

MoistureFraction = null

BwaDefaultRecordCount = count ***MonitorDefaultRecordsByHourLocation*** where ParameterCd = 'BWA'

If (*BwaDefaultRecordCount* is equal to 0)

MoistureFraction = 0.027

Else If (*BwaDefaultRecordCount* is equal to 1) AND (***MonitorDefaultRecordsByHourLocation*** record's DefaultValue is greater than 0 AND is less than 1)

MoistureFraction = ***MonitorDefaultRecordsByHourLocation*** record's DefaultValue

Else

return result F

If (***CalculationDiluent*** is null) OR (***CalculationConcentration*** is null) OR (***ValidFwFactorExists*** is false) OR (*MoistureFraction* is null)

return result C

Else if (***CalculationDiluent*** is equal to 20.9 * (1 - *MoistureFraction*))
return result D

Else if (***FinalConversionFactor*** is NOT null)
CalculatedUnadjustedValue = (***CalculationConversionFactor*** * ***CalculationConcentration*** * ***CurrentHourlyOpRecord***.FwFactor * [20.9 / (20.9 *(1 - *MoistureFraction*) - ***CalculationDiluent***)]) * ***FinalConversionFactor***

"19-3":
If ***CalculationConcentrationSubstituted*** OR ***CalculationDiluentSubstituted*** OR ***CalculationMoistureSubstituted***
return result A

Else if (***CalculationDiluent*** is null) OR (***CalculationConcentration*** is null) OR (***ValidFdFactorExists*** is false) OR (***CalculationMoisture*** is null)
return result C

Else if (***CalculationDiluent*** is equal to 20.9 * (100 - ***CalculationMoisture***) / 100)
return result D
Else if (***FinalConversionFactor*** is NOT null)
h2oFactor = (100 - ***CalculationMoisture***) / 100.0
denom = ((20.9 * *h2oFactor*) - ***CalculationDiluent***)

CalculatedUnadjustedValue = (***CalculationConversionFactor*** * ***CalculationConcentration*** * ***CurrentHourlyOpRecord***.FdFactor * (20.9 /*denom))** ***FinalConversionFactor***

"19-3D":
If ***CalculationConcentrationSubstituted*** OR ***CalculationDiluentSubstituted*** OR ***CalculationMoistureSubstituted***
return result A

Else if (***CalculationDiluent*** is null) OR (***CalculationConcentration*** is null) OR (***ValidFdFactorExists*** is false) OR (***CalculationMoisture*** is null)
return result C

Else if (***CalculationDiluent*** is equal to 20.9)
return result D

Else if (***FinalConversionFactor*** is NOT null)
h2oFactor = (100 - ***CalculationMoisture***) / 100.0
denom = (20.9 * *h2oFactor*) - (***CalculationDiluent*** * *h2oFactor*)

CalculatedUnadjustedValue = (***CalculationConversionFactor*** * ***CalculationConcentration*** * ***CurrentHourlyOpRecord***.FdFactor *(20.9 / *denom*)) * ***FinalConversionFactor***

"19-4":
If ***CalculationConcentrationSubstituted*** OR ***CalculationDiluentSubstituted*** OR ***CalculationMoistureSubstituted***
return result A

Else if (***CalculationDiluent*** is null) OR (***CalculationConcentration*** is null) OR (***ValidFdFactorExists*** is false) OR (***CalculationMoisture*** is null)
return result C

Else if (***CalculationDiluent*** is equal to 20.9) OR (***CalculationMoisture*** is equal to 100)
return result D

Else if (***FinalConversionFactor*** is NOT null)
CalculatedUnadjustedValue = (***CalculationConversionFactor*** * (***CalculationConcentration*** * ***CurrentHourlyOpRecord***.FdFactor / ((100 - ***CalculationMoisture***) / 100.0)) * (20.9 / (20.9 - ***CalculationDiluent***))) * ***FinalConversionFactor***

"19-5":
If ***CalculationConcentrationSubstituted*** OR ***CalculationDiluentSubstituted*** OR ***CalculationMoistureSubstituted***
return result A

Else if (***CalculationDiluent*** is null) OR (***CalculationConcentration*** is null) OR (***ValidFdFactorExists*** is false) OR (***CalculationMoisture*** is null)
return result C

Else if (***CalculationDiluent*** is equal to 20.9) OR (***CalculationMoisture*** is equal to 100)
return result D

Else if (***FinalConversionFactor*** is NOT null)
h2oFactor = (100 - ***CalculationMoisture***) / 100.0
denom = 20.9 - (***CalculationDiluent*** / *h2oFactor*)

CalculatedUnadjustedValue = (***CalculationConversionFactor*** * ***CalculationConcentration*** * ***CurrentHourlyOpRecord***.FdFactor * 20.9 / *denom)* * ***FinalConversionFactor***

"19-5D":
If ***CalculationConcentrationSubstituted*** OR ***CalculationDiluentSubstituted***
return result A

Else if (***CalculationDiluent*** is null) OR (***CalculationConcentration*** is null) OR (***ValidFdFactorExists*** is false)
return result C

Else if (***CalculationDiluent*** is equal to 20.9)
return result D

Else if (***FinalConversionFactor*** is NOT null)
CalculatedUnadjustedValue = (***CalculationConversionFactor*** * ***CalculationConcentration*** * ***CurrentHourlyOpRecord***.FdFactor * (20.9/ (20.9 - ***CalculationDiluent***))) * ***FinalConversionFactor***

"19-6" or "19-7":
If ***CalculationConcentrationSubstituted*** OR ***CalculationDiluentSubstituted***
return result A

Else if (***CalculationDiluent*** is null) OR (***CalculationConcentration*** is null) OR (***ValidFcFactorExists*** is false)

return result C

Else if (***CalculationDiluent*** is equal to 0.0)
return result D

Else if (***FinalConversionFactor*** is NOT null)
CalculatedUnadjustedValue = (***CalculationConversionFactor*** * ***CalculationConcentration*** * ***CurrentHourlyOpRecord***.FcFactor * (100.0 / ***CalculationDiluent***)) * ***FinalConversionFactor***

"19-8":
If ***CalculationConcentrationSubstituted*** OR ***CalculationDiluentSubstituted*** OR ***CalculationMoistureSubstituted***
return result A

Else if (***CalculationDiluent*** is null) OR (***CalculationConcentration*** is null) OR (***ValidFcFactorExists*** is false) OR (***CalculationMoisture*** is null)
return result C

Else if (***CalculationDiluent*** is equal to 0.0) OR (***CalculationMoisture*** is equal to 100)
return result D

Else if (***FinalConversionFactor*** is NOT null)
CalculatedUnadjustedValue = (***CalculationConversionFactor*** * ((***CalculationConcentration*** * ***CurrentHourlyOpRecord***.FcFactor) /((100 - ***CalculationMoisture***) / 100.0))* (100.0 / ***CalculationDiluent***)) * ***FinalConversionFactor***

"19-9":
If ***CalculationConcentrationSubstituted*** OR ***CalculationDiluentSubstituted*** OR ***CalculationMoistureSubstituted***
return result A

Else if (***CalculationDiluent*** is null) OR (***CalculationConcentration*** is null) OR (***ValidFcFactorExists*** is false) OR (***CalculationMoisture*** is null)
return result C

Else if (***CalculationDiluent*** is equal to 0.0)
return result D

Else if (***FinalConversionFactor*** is NOT null)
h2oFactor = (100 - ***CalculationMoisture***) / 100.0
co2Term = 100.0 / ***CalculationDiluent***

CalculatedUnadjustedValue = (***CalculationConversionFactor*** * ***CalculationConcentration*** * ***CurrentHourlyOpRecord***.FcFactor * *h2oFactor* * *co2Term)* * ***FinalConversionFactor***

Else
return result B

Results:

Result	Response	Severity
A	You reported an incorrect MODCCode in the MATS Derived Hourly Value record for [param], as you reported substitute data for one or more essential auxiliary parameters.	Critical Error Level 1
B	You reported an incorrect MODCCode in the MATS Derived Hourly Value record for [param], as you reported measured data for essential auxiliary parameters.	Critical Error Level 1
C	You reported a FormulaIdentifier in the MATS Derived Hourly Value record for [param], but you did not report a value for all the essential parameters needed to perform the calculation.	Critical Error Level 1
D	The [param] could not be recalculated, because the diluent value would result in division by zero.	Critical Error Level 1
F	You did not report a single valid MonitorDefault record for ParameterCode BWA for the hour.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Calculation Verification
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Calculation Verification
4	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Calculation Verification

Check Code: MATSCHV-16

Check Name: Stash Hg Calculated Value

Related Former Checks:

Applicability:

Description: Stores the HGRE or HGRH Calculated Unadjusted Value in the appropriate check parameters.

Specifications:

MatsCalculatedHgRateValue = ***CalculatedUnadjustedValue*** (converted to Scientific Notation, with 3 significant digits)

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Calculation Verification

Check Code: MATSCHV-17

Check Name: Stash HCl Calculated Value

Related Former Checks:

Applicability:

Description: Stores the HCLRE or HCLRH Calculated Unadjusted Value in the appropriate check parameters.

Specifications:

MatsCalculatedHclRateValue = ***CalculatedUnadjustedValue*** (converted to Scientific Notation, with 3 significant digits)

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Calculation Verification

Check Code: MATSCHV-18

Check Name: Stash HF Calculated Value

Related Former Checks:

Applicability:

Description: Stores the HFRE or HFRH Calculated Unadjusted Value in the appropriate check parameters.

Specifications:

MatsCalculatedHfRateValue = ***CalculatedUnadjustedValue*** (converted to Scientific Notation, with 3 significant digits)

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Calculation Verification

Check Code: MATSCHV-19

Check Name: Stash SO2 Calculated Value

Related Former Checks:

Applicability:

Description: Stores the SO2RE or SO2RH Calculated Unadjusted Value in the appropriate check parameters.

Specifications:

MatsCalculatedSo2RateValue = ***CalculatedUnadjustedValue*** (converted to Scientific Notation, with 3 significant digits)

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Calculation Verification

Check Code: MATSCHV-20

Check Name: Check Unadjusted Hourly Value Tolerance

Related Former Checks:

Applicability:

Description: Ensures that the percent difference between the Unadjusted Hourly Value and Calculated Unadjsuted Hourly Value is less than or equal to 5%.

Specifications:

If ***CurrentDhvRecordValid***

If (***MatsDhvRecord.***ModcCode in ***MatsDhvMeasuredModcList***) AND (***MatsDhvRecord***.UnadjustedHourlyValue is NOT null) AND (***CalculatedUnadjustedValue*** is NOT null)

If (***CurrentSo2MonitorHourlyRecord.***ParameterCode is not SO2RH and ***CurrentSo2MonitorHourlyRecord.***ParameterCode is not SO2RE) OR (***CurrentSo2MonitorHourlyRecord*** is null or ***CurrentSo2MonitorHourlyRecord.***ModcCode is not 16)

Set *roundedCalculatedValue* = ***CalculatedUnadjustedValue*** rounded to 3 significant digits

If ((***MatsDhvRecord***.UnadjustedHourlyValue + *roundedCalculatedValue*) is NOT equal to 0)

Set *PercentDifference* = 100 * ABS(***MatsDhvRecord***.UnadjustedHourlyValue - *roundedCalculatedValue*) / ((***MatsDhvRecord***.UnadjustedHourlyValue + *roundedCalculatedValue*) / 2), **rounded to 1 decimal place**.

If (*PercentDifference* > 5)
return result A

Results:

Result	Response	Severity
A	The UnadjustedHourlyValue reported in the DHV record for [param] is inconsistent with the recalculated value.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Calculation Verification
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Calculation Verification
3	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Calculation Verification
4	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Calculation Verification
5	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Calculation Verification
6	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Calculation Verification
7	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Calculation Verification
8	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Calculation Verification

Check Category:

MATS Derived Hourly Value Checks

Check Code: MATSDHV-1

Check Name: Initialize HGRE Derived Hourly Value Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for Hg

Specifications:

CurrentDhvParameter = "HGRE"
MatsDhvRecord = MatsHgDhvRecord
MatsEquationCodeWithH2o = "A-3"
MatsEquationCodeWithoutH2o = "A-2"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Derived Hourly Evaluation

Check Code: MATSDHV-2

Check Name: Initialize HGRH Derived Hourly Value Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for Hg

Specifications:

CurrentDhvParameter = "HGRH"
MatsDhvRecord = MatsHgDhvRecord

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Derived Hourly Evaluation

Check Code: MATSDHV-3

Check Name: Initialize HCLRE Derived Hourly Value Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for HCL

Specifications:

CurrentDhvParameter = "HCLRE"
MatsDhvRecord = MatsHclDhvRecord
MatsEquationCodeWithH2o = "HC-3"
MatsEquationCodeWithoutH2o = "HC-2"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Derived Hourly Evaluation

Check Code: MATSDHV-4

Check Name: Initialize HCLRH Derived Hourly Value Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for HCL

Specifications:

CurrentDhvParameter = "HCLRH"
MatsDhvRecord = MatsHclDhvRecord

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Derived Hourly Evaluation

Check Code: MATSDHV-5

Check Name: Initialize HFRE Derived Hourly Value Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for HF

Specifications:

CurrentDhvParameter = "HFRE"
MatsDhvRecord = MatsHfDhvRecord
MatsEquationCodeWithH2o = "HF-3"
MatsEquationCodeWithoutH2o = "HF-2"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Derived Hourly Evaluation

Check Code: MATSDHV-6

Check Name: Initialize HFRH Derived Hourly Value Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for HF

Specifications:

CurrentDhvParameter = "HFRH"
M***atsDhvRecord = MatsHfDhvRecord***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Derived Hourly Evaluation

Check Code: MATSDHV-7

Check Name: Initialize SO2RE Derived Hourly Value Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for SO2

Specifications:

CurrentDhvParameter = "SO2RE"
MatsDhvRecord = MatsSo2DhvRecord
MatsEquationCodeWithH2o = "S-3"
MatsEquationCodeWithoutH2o = "S-2"

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Derived Hourly Evaluation

Check Code: MATSDHV-8

Check Name: Initialize SO2RH Derived Hourly Value Data

Related Former Checks:

Applicability:

Description: This check sets generic parameters and output parameters for subsequent derived hourly checks for SO2

Specifications:

CurrentDhvParameter = "SO2RH"
M***atsDhvRecord = MatsSo2DhvRecord***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Derived Hourly Evaluation

Check Code: MATSDHV-9

Check Name: Check Mats MODC in DHV Records

Related Former Checks:

Applicability:

Description: Basic check to ensure that Mats MODC reported in the DHV record is valid.

Specifications:

DerivedHourlyModcStatus = false

If **Mats*DhvRecord.***ModcCode is equal to 36 or 38,
DerivedHourlyModcStatus = true

Else if **Mats*DhvRecord.***ModcCode is equal to 37,

If **Mats*DhvRecord.***ParameterCode is equal to "HGRH", "HCLRH", "HFRH" or "SO2RH", AND
CurrentHourlyOpRecord.MatsStartupShutdownFlag is NOT null,
DerivedHourlyModcStatus = true
Else
return result B

Else if **Mats*DhvRecord.***ModcCode is equal to 39,

If **Mats*DhvRecord.***ParameterCode is equal to "HGRE", "HCLRE", "HFRE" or "SO2RE", AND
CurrentHourlyOpRecord.MatsStartupShutdownFlag is NOT null,
DerivedHourlyModcStatus = true
Else
return result C

Else
return result A

Results:

Result	Response	Severity
A	You reported an MODC code that is not valid for the MATS DHV.	Critical Error Level 1
B	You reported MODC 37 for [PARAM], but did not report both a heat-input based MATS parameter, and a Startup/Shutdown Flag.	Informational Message
C	You reported MODC 39 for [PARAM], but did not report both an output based MATS parameter and a Startup/Shutdown Flag.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Derived Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Derived Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Derived Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Derived Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Derived Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Derived Hourly Evaluation

Check Code: MATSDHV-10

Check Name: Check Formula in MATS DHV Record

Related Former Checks:

Applicability:

Description: (Copy of DHV-24 tailored to MATS)
Checks the Formula ID in the MATS Derived Hourly Value record and ensures that it can be used for the calculation

Specifications:

DerivedHourlyFormulaStatus = false

If (***DerivedHourlyModcStatus*** = true)

If (***MatsDhvRecord***.FormulaKey is null)

If ***MatsDhvRecord***.ModcCode = "38"
return result G
else
return result A

else //FormulaKey not null

If (***MatsDhvRecord***.FormulaActiveInd is NOT equal to 1)
return result B

Else if (***MatsDhvRecord***.FormulaParameterCode is not equal to ***CurrentDhvParameter***)
return result C

Else if (***CurrentDhvParameter*** in set {"HGRE", "HCLRE", "HFRE", "SO2RE"} and ***MatsDhvRecord***.ModcCode = "37"
return result D

Else if (***CurrentDhvParameter*** in set {"HGRH", "HCLRH", "HFRH", "SO2RH"} and ***MatsDhvRecord***.ModcCode = "39"
return result E

Else
DerivedHourlyFormulaStatus = true

Results:

Result	Response	Severity
A	You did not report a FormulaID in the MATS DHV record for [param].	Critical Error Level 1
B	You reported FormulaID [ID] in the MATS DHV record for [param], but there is no active Formula record for this formula in your monitoring plan.	Critical Error Level 1
C	You reported FormulaID [ID] in the MATS DHV record for [param], but in your monitoring plan this formula has a different ParameterCode.	Critical Error Level 1
D	You reported a MODCCode of 37 for the MATS DHV record, but the use of a diluent cap value is not applicable to [param].	Critical Error Level 1
E	You reported a MODCCode of 39 for the MATS DHV record, but the use of a default electrical load value is not applicable to [param].	Critical Error Level 1
F	You reported a FormulaID for a MATS DHV record, that is not reported if valid concentration was not available or substitute data reported for one or more essential auxiliary parameters.	Critical Error Level 1
G	You did not report a FormulaID in the MATS DHV record for [param].	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Derived Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Derived Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Derived Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Derived Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Derived Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Derived Hourly Evaluation

Check Code: MATSDHV-11

Check Name: Check Equation Code for MATS RE

Related Former Checks:

Applicability:

Description: Gets Equation Code from Mats Active Monitor Formula Record and verifies that it is an appropriate equation for calculation of HCLRE,HFRE,HGRE,SO2RE

Specifications:

DerivedHourlyEquationStatus = false

If (***DerivedHourlyFormulaStatus*** == true)

If (***MatsDhvRecord***.EquationCode is not null)

If (***MatsDhvRecord***.EquationCode ==***MatsEquationCodeWithoutH2o***)

DerivedHourlyEquationStatus = true
FlowMonitorHourlyChecksNeeded = true

Else If (***MatsDhvRecord***.EquationCode ==***MatsEquationCodeWithH2o***)

Derived Hourly Equation Status = true
Flow Monitor Hourly Checks Needed = true
Moisture Needed = true
append "MIN" to ***H2OMissingDataApproach***

Else
return result A

Else
DerivedHourlyEquationStatus = true

Results:

Result	Response	Severity
A	You reported a formula code that does not match any of the MATS derived hourly value formulas.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Derived Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Derived Hourly Evaluation

Check Code: MATSDHV-12

Check Name: Check Equation Code for MATS RH

Related Former Checks:

Applicability:

Description: Gets Mats Equation Code from Active Mats Monitor Formula Record and verifies that it is an appropriate equation for Mats Current parameter.

Specifications:

DerivedHourlyEquationStatus = false

If (***DerivedHourlyFormulaStatus*** == true)

If (***MatsDhvRecord***.EquationCode is not null)

If (***MatsDhvRecord***.EquationCode in set {19-1, 19-2, 19-3, 19-3D, 19-4, 19-5, 19-5D, 19-6, 19-7, 19-8, 19-9})

DerivedHourlyEquationStatus = true

If (***MatsDhvRecord***.EquationCode in set {19-1, 19-4})
O2DryNeededForMats = true
FDFactorNeeded = true

Else if (***MatsDhvRecord***.EquationCode in set {19-3,19-3D, 19-5, 19-5D})
O2WetNeededForMats = true
FDFactorNeeded = true

Else if (***MatsDhvRecord***.EquationCode in set {19-2})
O2WetNeededForMats = true
FWFactorNeeded = true

Else if (***MatsDhvRecord***.EquationCode in set {19-6, 19-7, 19-8, 19-9})
CO2DiluentNeededForMats = true
FCFactorNeeded = true

If (***MatsDhvRecord***.EquationCode in set {19-3, 19-3D, 19-4, 19-5, 19-8, 19-9}
MoistureNeeded = true

Else
return result A

Else
DerivedHourlyEquationStatus= true

Results:

Result	Response	Severity
A	You reported a formula code that does not match any of the MATS derived hourly value formulas.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Derived Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Derived Hourly Evaluation

Check Code: MATSDHV-13

Check Name: Complete HGRE and HGRH Derived Hourly Value

Related Former Checks:

Applicability:

Description: This check assigns parameter specific check parameters used by the associated calculation checks.

Specifications:

MatsHgDhvParameter = ***CurrentDhvParameter***
MatsHgDhvValid = ***DerivedHourlyEquationStatus*** AND ***DerivedHourlyUnadjustedValueStatus***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Derived Hourly Evaluation

Check Code: MATSDHV-14

Check Name: Complete HCLRE and HCLRH Derived Houly Value

Related Former Checks:

Applicability:

Description: This check assigns parameter specific check parameters used by the associated calculation checks.

Specifications:

MatsHclDhvParameter = ***CurrentDhvParameter***
MatsHclDhvValid = ***DerivedHourlyEquationStatus*** AND ***DerivedHourlyUnadjustedValueStatus***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Derived Hourly Evaluation

Check Code: MATSDHV-15

Check Name: Complete HFRE and HFRH Derived Hourly Value

Related Former Checks:

Applicability:

Description: This check assigns parameter specific check parameters used by the associated calculation checks.

Specifications:

MatsHfDhvParameter = ***CurrentDhvParameter***
MatsHfDhvValid = ***DerivedHourlyEquationStatus*** AND ***DerivedHourlyUnadjustedValueStatus***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Derived Hourly Evaluation

Check Code: MATSDHV-16

Check Name: Complete SO2RE and SO2RH Derived Hourly Value

Related Former Checks:

Applicability:

Description: This check assigns parameter specific check parameters used by the associated calculation checks.

Specifications:

MatsSo2DhvParameter = ***CurrentDhvParameter***
MatsSo2DhvValid = ***DerivedHourlyEquationStatus*** AND ***DerivedHourlyUnadjustedValueStatus***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Derived Hourly Evaluation

Check Code: MATSDHV-17

Check Name: Check Unadjusted Value

Related Former Checks:

Applicability:

Description:

Specifications:

DerivedHourlyUnadjustedValueStatus = false

If (***DerivedHourlyModcStatus*** = true)

If (***MatsDhvRecord***.ModcCode in set {36, 37, 39})

If (***MatsDhvRecord***.UnadjustedHourlyValue is null)
return result A

Else if (***MatsDhvRecord***.UnadjustedHourlyValue is NOT reported in scientific notation to three significant digits)
return result B

Else if (***MatsDhvRecord***.UnadjustedHourlyValue < 0)
return result C

Else
DerivedHourlyUnadjustedValueStatus = true

Else // MODC 38

If (***MatsDhvRecord***.UnadjustedHourlyValue is not null)
return result D

Else
DerivedHourlyUnadjustedValueStatus = true

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1
B	The [fieldname] value in the [key] records is not reported in scientific notation rounded to three significant figures, with one digit to the left of the decimal point.	Critical Error Level 1
C	You reported a negative value, which is invalid, in the field [fieldname] for [key].	Critical Error Level 1
D	You reported an UnadjustedHourlyValue for a MATS DHV record, that is not reported if a valid concentration was not available or substitute data reported for one or more essential auxiliary parameters.	Critical Error Level 2

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Derived Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Derived Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Derived Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Derived Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Derived Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Derived Hourly Evaluation

Check Code: MATSDHV-18

Check Name: Set Diluents Needed for Calculation

Related Former Checks:

Applicability: General Check

Description: Uses measured MODC list and CO2 Diluent, O2 Dry and O2 Wet Needed for MATS check parameters to set corresponding Needed for MATS Calculation check parameters.

Specifications:

If (***DerivedHourlyEquationStatus*** = true) AND (***DerivedHourlyModcStatus*** = true) AND (***MatsDhvRecord***.ModcCode set {36, 37, 39})

If (***CO2DiluentNeededForMats*** = true)
CO2DiluentNeededForMatsCalculation = true

If (***O2DryNeededForMats*** = true)
O2DryNeededForMatsCalculation = true

If (***O2WetNeededForMats*** = true)
O2WetNeededForMatsCalculation = true

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RE Derived Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl RH Derived Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RE Derived Hourly Evaluation
4	Process/Category:	Emissions Data Evaluation Report ------ MATS HF RH Derived Hourly Evaluation
5	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RE Derived Hourly Evaluation
6	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg RH Derived Hourly Evaluation
7	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RE Derived Hourly Evaluation
8	Process/Category:	Emissions Data Evaluation Report ------ MATS SO2 RH Derived Hourly Evaluation

Check Category:

MATS Hourly GFM Data

Check Code: MATSGFM-1

Check Name: Component ID Valid

Related Former Checks:

Applicability:

Description: Ensure that the Component ID is associated with “STRAIN”.

Specifications:

Set ***MatsGfmSamplingTrainRecords*** to null.
Set ***MatsHourlyGFMComponentIdValid*** = false.
Set ***MatsSamplingTrainRecord*** to null.
Set ***MatsSamplingTrainQaStatus*** to null.
Set ***MatsSorbentTrapBeginDateHour*** to null.
Set ***MatsSorbentTrapEndDateHour*** to null.
Set ***MatsSamplingTrainCount*** to null.

If the ***MatsHourlyGFMRecord***.ComponentID is null,

Return result A.

Else

Locate ***MatsSamplingTrainRecords*** where:

1) ComponentId is equal to ***MatsHourlyGFMRecord***.ComponentID
2) SorbentTrapBeginDateHour is on or before ***CurrentDateHour***
3) SorbentTrapEndDateHour is on or after ***CurrentDateHour***
4) Records are sorted by SorbentTrapBeginDateHour and SorbentTrapEndDateHour // The earliest sampling train is the correct train.

Set ***MatsSamplingTrainCount*** to the number of records located in ***MatsSamplingTrainsRecords***.
Set ***MatsGfmSamplingTrainRecords*** to the records located in ***MatsSamplingTrainsRecords***.

If not found,

Return result B.

Else

Set ***MatsHourlyGFMComponentIdValid*** to true.
Set ***MatsSamplingTrainRecord*** to the first record located in ***MatsSamplingTrainsRecords***.
Set ***MatsSamplingTrainQaStatus*** to ***MatsSamplingTrainRecord***.TrainQAStatusCode.
Set ***MatsSorbentTrapBeginDateHour*** to ***MatsSamplingTrainRecord***.BeginDateHour.
Set ***MatsSorbentTrapEndDateHour*** to ***MatsSamplingTrainRecord***.EndDateHour.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	For [key], you reported a sorbent train GFM Component ID that does not match a sorbent train Component ID record.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ MATS Hourly Gas Flow Meter Evaluation

Check Code: MATSGFM-2

Check Name: Begin and End Hour Flags Valid

Related Former Checks:

Applicability:

Description: Check that Begin and End Hour Flags are valid

Specifications:

If ***MatsHourlyGFMComponentIdValid*** is true,

If ***HourlyGFMData***.BeginEndHourFlag is "I",

If ***CurrentDateHour*** is not equal to the ***MatsSorbentTrapBeginDateHour*** and is not the hour after the ***MatsSorbentTrapBeginDateHour***.

Return result A

Else if the ***HourlyGFMData***.BeginEndHourFlag is "F",

If ***CurrentDateHour*** is not equal to the***MatsSorbentTrapEndDateHour*** and is not the hour before the ***MatsSorbentTrapEndDateHour***,

Return result B.

Else if ***HourlyGFMData***.BeginEndHourFlag is null,

If ***CurrentDateHour*** is on the ***MatsSorbentTrapBeginDateHour,***

Return result C.

Else if ***CurrentDateHour*** is on the ***MatsSorbentTrapEndDateHour***,

Return result D.

Else if ***HourlyGFMData***.BeginEndHourFlag is "T",

If ***MatsSamplingTrainCount*** is less than or equal to 1,
Return result E.

Else if ***CurrentDateHour*** is not equal to the ***MatsSorbentTrapEndDateHour***

// Current hour is not the end hour of the current sorbent trap
Return result F.

Results:

Result	Response	Severity
A	For [key], you identified a begin hour that is not the first or second hour of the sampling period.	Critical Error Level 1
B	For [key], you identified an end hour that is not the last or second to the last hour of the sampling period.	Critical Error Level 1
C	For [key], you did not identify the first hour of the sampling period as a begin or transition hour.	Critical Error Level 1
D	For [key], you did not identify the last hour of the sampling period as an end or transition hour.	Critical Error Level 1
E	For [key], you identified an hour as a transition hour, but the hour is not included in two consecutive sampling periods.	Critical Error Level 1
F	For [key], you identified a transition hour that is not the last hour of a sampling period.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ MATS Hourly Gas Flow Meter Evaluation

Check Code: MATSGFM-3

Check Name: Hourly GFM Reading Valid

Related Former Checks:

Applicability:

Description: Hourly GFM Reading Null or Reported to Two Decimal Places

Specifications:

If ***MatsHourlyGFMComponentIdValid*** is true,

If the ***MatsHourlyGFMRecord***.HourlyGFMReading is null,

If the ***MatsSamplingTrainQaStatus*** is NOT equal to "INC", "EXPIRED", or "LOST", AND ***HourlyGFMData***.BeginEndHourFlag is NOT equal to 'N',

Return result A.

Else,

If ***HourlyGFMData***.BeginEndHourFlag is equal to 'N',

Return result D.

Else if the ***MatsSamplingTrainQaStatus*** is not "PASSED", "FAILED", or "UNCERTAIN",

Return result B.

Else if the ***MatsHourlyGFMRecord***.HourlyGFMReading is not reported to two decimal places ,

Return result C.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	For [key], the [fieldname] value must be reported to at least two decimal places.	Critical Error Level 1
D	You reported a [fieldname] value in the [key] records, but also reported a problem with the hourly GFM data with a "N" in the BeginEndHourFlag record.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ MATS Hourly Gas Flow Meter Evaluation

Check Code: MATSGFM-4

Check Name: Average Hourly Sampling Rate Valid

Related Former Checks:

Applicability:

Description: Average Hourly Sampling Rate Null or Reported to Two Decimal Places

Specifications:

If ***MatsHourlyGFMComponentIdValid*** is true,

If the ***MatsHourlyGFMRecord***.AvgHourlySamplingRate is null,

If the ***MatsSamplingTrainQaStatus*** is NOT equal to "INC", "EXPIRED", or "LOST", AND ***HourlyGFMData***.BeginEndHourFlag is NOT equal to 'N',

Return result A.

Else,

If ***HourlyGFMData***.BeginEndHourFlag is equal to 'N',

Return result D.

Else if the ***MatsSamplingTrainQaStatus*** is not "PASSED", "FAILED", or "UNCERTAIN",

Return result B.

Else if the ***MatsHourlyGFMRecord***.AvgHourlySamplingRate is not reported to two decimal places ,

Return result C.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	For [key], the [fieldname] value must be reported to at least two decimal places.	Critical Error Level 1
D	You reported a [fieldname] value in the [key] records, but also reported a problem with the hourly GFM data with a "N" in the BeginEndHourFlag record.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ MATS Hourly Gas Flow Meter Evaluation

Check Code: MATSGFM-5

Check Name: Sampling Rate UOM Valid

Related Former Checks:

Applicability:

Description: Sampling Rate UOM Null or Matches UOM Code

Specifications:

If ***MatsHourlyGFMComponentIdValid*** is true,
If the ***MatsHourlyGFMRecord***.SamplingRateUOM is null,

If the ***MatsSamplingTrainQaStatus*** is NOT equal to "INC", "EXPIRED", or "LOST", AND ***HourlyGFMData***.BeginEndHourFlag is NOT equal to 'N',

Return result A.

Else,

If ***HourlyGFMData***.BeginEndHourFlag is equal to 'N',

Return result D.

Else if the ***MatsSamplingTrainQaStatus*** is not "PASSED", "FAILED", or "UNCERTAIN",

Return result B.

Else if the ***MatsHourlyGFMRecord***.SamplingRateUOM is not "CCMIN", "DSCMMIN", "LMIN", "CCHR", "DSCMHR", or "LHR" ,

Return result C.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	For [key] you reported a [value] which is not valid for [fieldname].	Critical Error Level 1
D	You reported a [fieldname] value in the [key] records, but also reported a problem with the hourly GFM data with a "N" in the BeginEndHourFlag record.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ MATS Hourly Gas Flow Meter Evaluation

Check Code: MATSGFM-6

Check Name: Hourly SFSR Ratio Valid

Related Former Checks:

Applicability:

Description: Hourly SFSR Ratio Null or Reported in Required Range and Decimal Places

Specifications:

If ***MatsHourlyGFMComponentIdValid*** is true,

If the ***MatsHourlyGFMRecord***.HourlySFSRRatio is null,

If ***MatsSamplingTrainQaStatus*** is not "INC", "EXPIRED", "LOST" or "FAILED", AND ***CurrentStackFlowHourlyRecord*** is NOT null AND ***CurrentStackFlowHourlyRecord***.ModcCode in set {01, 02, 03, 04, 20, 53, 54}, AND ***HourlyGFMData***.BeginEndHourFlag is NOT equal to 'N',

Return result A.

Else,

If ***HourlyGFMData***.BeginEndHourFlag is equal to 'N',

Return result G.

Else if ***MatsSamplingTrainQaStatus*** is not "PASSED", "FAILED", or "UNCERTAIN",

Return result B.

Else if ***CurrentStackFlowHourlyRecord*** is null,

Return result F.

Else if ***CurrentStackFlowHourlyRecord***.ModcCode NOT in set {01, 02, 03, 04, 20, 53, 54},

Return result E.

Else if the the ***MatsHourlyGFMRecord***.HourlySFSRRatio is not reported to one decimal place ,

Return result C.

Else if the ***MatsHourlyGFMRecord.***HourlySFSRRatio is not greater than or equal to 1.0 and less than or equal to 100.0,

Return result D.

Else if ***MatsHourlyGFMComponentIdValid*** is true AND ***MatsSamplingTrainDictionary*** contains a key equal to ***MatsSamplingTrainRecord***.TrainID AND ***MatsSamplingTrainDictionary***.ReferenceSFSRRatio where the key equals ***MatsSamplingTrainRecord***.TrainID is NOT null or equal to 0,

Set ***MatsHourlySfsrRatioDeviation*** to absolute value of [1 - (***MatsHourlyGFMRecord***.HourlySFSRRatio / ***MatsSamplingTrainDictionary***.ReferenceSFSRRatio where the key equals ***MatsSamplingTrainRecord***.TrainID)] x 100, rounded to an integer.

Add one to ***MatsSamplingTrainDictionary***.TotalSFSRRatioCount where the key equals ***MatsSamplingTrainRecord***.TrainID

If the ***MatsHourlySfsrRatioDeviation*** is greater than 25,

Add one to ***MatsSamplingTrainDictionary***.DeviatedSFSRRatioCount where the key equals ***MatsSamplingTrainRecord***.TrainID

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	The [fieldname] value for [key] should be reported to one decimal place.	Critical Error Level 1
D	The [fieldname] value for [key] must be a number between 1 and 100.	Critical Error Level 1
E	You reported a [fieldname] value in the [key] records which is not reported if the stack gas flow rate for the hour is a substitute data value.	Critical Error Level 1
F	You reported a [fieldname] value in the [key] records which you should not report when a stack gas flow rate is not reported for the hour.	Critical Error Level 1
G	You reported a [fieldname] value in the [key] records, but also reported a problem with the hourly GFM data with a "N" in the BeginEndHourFlag record.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ MATS Hourly Gas Flow Meter Evaluation

Check Code: MATSGFM-7

Check Name: Count Begin and End Hour Flags

Related Former Checks:

Applicability:

Description: Updates the total hours that a GFM exists for a sampling train and the count of hours where the Begin and End Flag equals "N".

Specifications:

If ***MatsGfmSamplingTrainRecords*** is not null,

For each *SamplingTrainRecord* in ***MatsGfmSamplingTrainRecords***,

If *SamplingTrainRecord*.TrainQAStatusCode is equal to "PASSED", "FAILED" or "UNCERTAIN", AND *SamplingTrainRecord*.RataInd is equal to 0 (zero) or null,

If ***MatsSamplingTrainDictionary*** contains a key equal to *SamplingTrainRecord*.TrainID,

Add one to ***MatsSamplingTrainDictionary***.TotalGfmCount where the key equals *SamplingTrainRecord*.TrainID

If ***MatsHourlyGFMData***.BeginEndHourFlag is equal to 'N',

Add one to ***MatsSamplingTrainDictionary***.NotAvailableGfmCount where the key equals *SamplingTrainRecord*.TrainID

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Hourly Gas Flow Meter Evaluation

Check Category:

MATS Monitor Hourly Value Checks

Check Code: MATSMHV-1

Check Name: MATS HgC: Initialize

Related Former Checks:

Applicability:

Description:

Specifications:

Set ***CurrentMhvParameter*** to "HGC"
Set ***MatsMhvRecord*** to ***MatsHgcMhvRecord***
Set ***MatsMhvSorbentTraps*** to null
Set ***MatsMhvSupplementalSorbentTraps*** to null

If (***MatsHgMethodRecord***.MethodCode is equal to "ST") OR ((***MatsHgMethodRecord***.MethodCode is equal to "CEMST") AND (***MatsHgcMhvRecord***.SystemTypeCode is equal to "ST"))

Set ***CurrentMhvSystemType*** to "ST"
Set ***CurrentMhvComponentType*** to "STRAIN"
Set ***MatsMhvMeasuredModcList*** to {01, 02, 32, 33, 41, 42, 43, 44}

Locate ***MatsSorbentTrapRecords*** where:

1) SystemId is equal to ***MatsMhvRecord***.SystemID
2) BeginDateHour is on or before ***CurrentDateHour***
3) EndDateHour is on or after ***CurrentDateHour***

Set ***MatsMhvSorbentTraps*** to the located records

Else

Set ***CurrentMhvSystemType*** to "HG"
Set ***CurrentMhvComponentType*** to "HG"
Set ***MatsMhvMeasuredModcList*** to {01, 02, 17, and 21}

Set ***MatsMhvUnavailableModcList*** to {34 and 35}
Set ***MatsMhvNoLikeKindModcList*** to {01 and 02}

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-2

Check Name: MATS HClC: Initialize

Related Former Checks:

Applicability:

Description:

Specifications:

Set ***CurrentMhvParameter*** to "HCLC"

Set ***MatsMhvRecord*** to ***MatsHclcMhvRecord***
Set ***CurrentMhvComponentType*** to "HCL"
Set ***CurrentMhvSystemType*** to"HCL"

Set ***MatsMhvMeasuredModcList*** to {01, 02, 17, and 21}
Set ***MatsMhvUnavailableModcList*** to {34 and 35}
Set ***MatsMhvNoLikeKindModcList*** to {01 and 02}

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-3

Check Name: MATS HFC: Initialize

Related Former Checks:

Applicability:

Description:

Specifications:

Set ***CurrentMhvParameter*** to "HFC"

Set ***MatsMhvRecord*** to ***MatsHfcMhvRecord***
Set ***CurrentMhvComponentType*** to "HF"
Set ***CurrentMhvSystemType*** to "HF"

Set ***MatsMhvMeasuredModcList*** to {01, 02, 17 and 21}
Set ***MatsMhvUnavailableModcList*** to {34 and 35}
Set ***MatsMhvNoLikeKindModcList*** to {01 and 02}

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-4

Check Name: MATS: Check MODC

Related Former Checks:

Applicability:

Description: Ensure that the reported MODC is one of the valid measured or unavailable MODC for the MATS parameter.

Specifications:

Set ***MonitorHourlyModcStatus*** to false

If (***MatsMhvRecord***.ModcCode is null)
return result A

Else if (***MatsMhvRecord***.ModcCode not in ***MatsMhvMeasuredModcList*** AND not in ***MatsMhvUnavailableModcList***)
return result B

Else if (***CurrentMhvSystemType*** is equal to "ST") AND (***MatsMhvRecord***.ModcCode is NOT equal to "41" or "42")

If (***MatsMhvSorbentTraps*** count is greater than 0)

Set *MatchingSorbentTrapsFound* to false.

Set *MatchingSorbentTrapsFound* to true if a ***MatsMhvSorbentTraps*** record exists where:

1) ModcCode is equal to ***MatsMhvRecord***.ModcCode
2) HgConcentration is equal to ***MatsMhvRecord***.UnadjustedValue

If (*MatchingSorbentTrapsFound* is false) AND (***MatsMhvRecord***.ModcCode is equal to "35")

Set *MatchingSorbentTrapsFound* to true if a ***MatsMhvSorbentTraps*** record exists where:

1) ModcCode is equal to "01" or "02".

If (*MatchingSorbentTrapsFound* is true)
MonitorHourlyModcStatus = true
Else
return result D

Else
MonitorHourlyModcStatus = true

Else
MonitorHourlyModcStatus = true

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1
B	The MODCCode reported for MATS Monitor Hourly Value is invalid.	Critical Error Level 1
C	The MODCCode reported for MATS Monitor Hourly Value is invalid for Hg sorbent trap systems.	Critical Error Level 1
D	The MODCCode and UnadjustedValue combination reported for MATS Monitor Hourly Value do not match the values for a reported Hg sorbent trap systems.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-5

Check Name: MATS: Check Percent Monitor Availability

Related Former Checks:

Applicability:

Description: Ensures that the Percent Monitor Availability (PMA) was reported and is inclusively between 0 and 100.

Specifications:

Set ***MonitorHourlyPmaStatus*** = false

If (***MonitorHourlyModcStatus*** == true)

If (***MatsMhvRecord***.PercentAvailable is NULL)

return result A

Else if (***MatsMhvRecord***.PercentAvailable> 100.0 OR ***MatsMhvRecord***.PercentAvailable < 0.0)

return result B

Else

Set ***MonitorHourlyPmaStatus*** = true

Results:

Result	Response	Severity
A	You did not report a PMA for the MATS Monitor Hourly Value.	Critical Error Level 1
B	The PMA must be in a range from 0 to 100.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-6

Check Name: MATS: Monitoring System

Related Former Checks:

Applicability:

Description: Ensures that a Monitor System is reported when a measured MODC is reported, and that is not reported when an unavailable MODC is reported. When Monitor System is and should have been reported, the check ensures that the system type is valid for the MATS parameter being reported.

Specifications:

Set ***MonitorHourlySystemStatus*** = false

If (***MonitorHourlyModcStatus*** == true)

If ***MatsMhvRecord***.MonitoringSystemID is null
If (***MatsMhvRecord.***ModcCode in ***MatsMhvMeasuredModcList)***
return result A
Else
return result F

Else if ***MatsMhvRecord***.SystemIdentifier is null
return result B

Else if (***MatsMhvRecord***.SystemTypeCode <> ***CurrentMhvSystemType***)
return result C

Else if (***MatsMhvRecord***.SystemTypeCode is equal to "ST")

If ***MatsMhvSorbentTraps*** count is greater than 0,
MonitorHourlySystemStatus = true
Else
If (***MatsMhvRecord.***ModcCode in ***MatsMhvMeasuredModcList)***
return result E

Else
MonitorHourlySystemStatus = true

Results:

Result	Response	Severity
A	You did not report a MonitoringSystemID for the [param] MATS Monitor Hourly Value. MonitoringSystemID is required when you report measured data.	Critical Error Level 1
B	You reported MonitoringSystemID [ID] in the MATS MHV record for [param], but there is no Monitoring System record for this system in your monitoring plan that was active during the hour.	Critical Error Level 1
C	The system type associated with the Monitoring System ID for the [param] MATS Monitor Hourly Value is not consistent with that parameter.	Critical Error Level 1
D	You reported a MonitoringSystemID for the [param] MATS Monitor Hourly Value that is not reported based on the MODCCode.	Critical Error Level 1
E	You reported sorbent trap system [ID] in the MATS MHV record for [param], but the emissions report does not contain a sorbent trap record for the system that was active during the current hour.	Critical Error Level 1
F	You did not report a MonitoringSystemID for the [param] MATS Monitor Hourly Value. MonitoringSystemID is required when you report unavailable data.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-7

Check Name: MATS: System Designation

Related Former Checks:

Applicability:

Description: Ensure that the System Designation Code is valid for the reported MODC.

Currently, the system designation should be 'P' for MODC 01 and 17, 'B' or 'PB' for MODC 02.

Specifications:

If (***MonitorHourlyModcStatus*** == true AND ***MonitorHourlySystemStatus*** == true AND ***MatsMhvRecord***.SystemIdentifier is not null)

case (***MatsMhvRecord.***ModcCode)

01 OR 17: If (***MatsMhvRecord***.SystemDesignationCode <> "P")
return result A

02: If (***MatsMhvRecord***.SystemDesignationCode NOT in set {B, RB}
return result B

Results:

Result	Response	Severity
A	You reported a MATS Hourly Value MODCCode that is only used with primary or temporary like kind monitoring systems.	Critical Error Level 1
B	You reported a MATS Monitor Hourly Value MODCCode that is only used with backup or redundant backup monitoring systems.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-8

Check Name: MATS: Check Like Kind Analyzer Use

Related Former Checks:

Applicability:

Description: Ensures that the conditions exist that allow the use of particular MODC.

Currently only checks MODC 17.

Specifications:

If (***MonitorHourlyModcStatus*** == true AND ***MonitorHourlySystemStatus*** == true)

If (***MatsMhvRecord***.ModcCode == "17")

HoursOfUseOfLikeKindAnalyzer = Count of MonitorHourlyValueData records for the location and reporting period where:

1) ParameterCode = ***CurrentMhvParameter***
2) ModcCode == "17"
3) BeginDateHour < ***CurrentOperatingDateHour***

If *HoursOfUseOfLikeKindAnalyzer* >= 720

FirstUseOfLikeKindAnalyzerRecord = MonitorHourlyValueData record at earliest time for the location and reporting period where:

1) ParameterCode = ***CurrentMhvParameter***
2) ModcCode == "17"
3) BeginDateHour < ***CurrentOperatingDateHour***

Locate a ***RATATestRecordsByLocationForQAStatus*** for the location where:

1) MonitoringSystemID is equal to ***MatsMhvRecord***.MonitoringSystemID
2) TestResultCode begins with "PASS"
3) EndDate/EndHour is after the *FirstUseOfLikeKindAnalyzerRecord*.Date/Hour and on or prior to the ***CurrentOperatingDateHour***.

If not found,
return result A

Results:

Result	Response	Severity
A	You reported an MODCCode of 17 in the MATS Monitor Hourly Value record for [param], indicating the use of a like-kind analyzer, but you have used a like-kind analyzer to monitor this parameter for more than 720 hours during this reporting period. You are not allowed to use a like-kind analyzer for more than 720 hours during a calendar year, unless the analyzer is identified as a non-redundant backup and a RATA is performed.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-9

Check Name: MATS: Component

Related Former Checks:

Applicability:

Description: Ensures that a Component is reported when a measured MODC is reported, and that is not reported when an unavailable MODC is reported. When Component is and should have been reported, the check ensures that the system type is valid for the MATS parameter being reported.

Specifications:

MonitorHourlyComponentStatus = false

If (***CurrentMhvSystemType*** <> "ST")

If (***MonitorHourlyModcStatus*** = true)

If ***MatsMhvRecord.***ComponentID is null

If (***MatsMhvRecord***.ModcCode in set ***MatsMhvMeasuredModcList***)
return result A
Else
return result G

Else if ***MatsMhvRecord***.ComponentIdentifier is null
return result B

Else if (***MatsMhvRecord***.ComponentTypeCode <> ***CurrentMhvComponentType***)
return result C

Else if ***MatsMhvRecord.***ModcCode == 17 AND ***MatsMhvRecord***.ComponentIdentifier does not begin with "LK"
return result D

Else if ***MatsMhvRecord***.ComponentIdentifier begins with "LK" AND ***MatsMhvRecord.***ModcCode in ***MatsMhvNoLikeKindModcList***
return result H

Else
MonitorHourlyComponentStatus = true

Else

If ***MatsMhvRecord.***ComponentID is NOT null
return result F

Else
MonitorHourlyComponentStatus = true

Results:

Result	Response	Severity
A	You did not report a ComponentID for the [param] MATS Monitor Hourly Value.	Critical Error Level 1
B	Your reported ComponentID [ID] in the MATS MHV record for [param], but there is no Component record for this component in your monitoring plan.	Critical Error Level 1
C	The component type associated with the ComponentID for the [param] MATS Monitor Hourly Value is not consistent with the parameter.	Critical Error Level 1
D	You reported an MODCCode of 17 in the MATS MHV record for [param], which indicates that the component is a like-kind analyzer, but the ComponentID does not begin with LK.	Critical Error Level 1
E	You reported a ComponentID for the [param] MATS Monitor Hourly Value that is not reported based on the MODCCode.	Critical Error Level 1
F	You reported a ComponentID in the MATS MHV record for a [type] system, but a ComponentID is only reported for a gas CEMS.	Critical Error Level 1
G	You did not report a ComponentID for the [param] MATS Monitor Hourly Value.	Critical Error Level 1
H	You reported a ComponentID in the [param] MHV record that begins with "LK", but did not report an MODCCode of 17. You must report an MODCCode of 17 when a like-kind analyzer is used.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-10

Check Name: MATS: System Component

Related Former Checks:

Applicability:

Description: Ensure that at least one active Monitoring System Component record exists for the Monitoring System Id and Component Id in the current MATS MHV record.

Specifications:

If (***MonitorHourlySystemStatus*** == true) AND (***MatsMhvRecord***.MonitoringSystemID is not null) AND (***MonitorHourlyComponentStatus*** == true) AND (***MatsMhvRecord***.ComponentID is not null)

CountMonSysCompRecord = count ***MonitoringSystemComponentByHourLocation*** records where:

1) MonitoringSystemID = ***MatsMhvRecord.***MonitoringSystemID
2) ComponentID = ***MatsMhvRecord***.ComponentID

If *CountMonSysCompRecord* = 0
return result A

Results:

Result	Response	Severity
A	You reported MonitoringSystemID [sys] ComponentID [ID] in the MATS MHV record for [param], but there is no MonitorSystemComponent record for this system and component in your monitoring plan that was active during the hour.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-11

Check Name: MATS: Max/Min Value

Related Former Checks:

Applicability:

Description: Determines the MPC for the active Monitor Span record for the hour, location and component type, returning a check result if the a single row is not found or the MPC is not greater than 0.

Specifications:

CurrentMHVMaxMinValue = null

If (***MonitorHourlyModcStatus*** == true AND ***MatsMhvRecord***.ModcCode in set ***MatsMhvMeasuredModcList***)

If (***CurrentMhvComponentType*** == "HG")

MonitorSpanRecordCount = Find active ***MonitorSpanRecordByHourAndLocation*** where:

1) ComponentTypeCode = ***CurrentMhvComponentType*** AND
2) SpanScaleCode = "H"

If (*MonitorSpanRecordCount* > 1)
return result A

Else if (*MonitorSpanRecordCount* = 0)
return result B

Else
CurrentMonitorSpanRecord = the single matched record

If *CurrentMonitorSpanRecord*.MPCValue > 0)

CurrentMhvMaxMinValue = *CurrentMonitorSpanRecord*.MPCValue

Else
return result C

Results:

Result	Response	Severity
A	You have more than one active High Range SpanScaleCode at the current location for the hour.	Critical Error Level 1
B	You have no active High Range SpanScaleCode at the current location for the hour.	Critical Error Level 1
C	The value in the reported span record for [param] is invalid.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-12

Check Name: MATS: Unadjusted Value

Related Former Checks:

Applicability:

Description: Validates the value and format of the reported unadjusted hourly value.

Specifications:

MonitorHourlyUnadjustedValueStatus = false
MatsMhvCalculatedValue = null

If (***MonitorHourlyModcStatus*** == true)

Case (***MatsMhvRecord.***ModcCode)

= 21:

MatsMhvCalculatedValue = 0.00E0

If (***MatsMhvRecord***.UnadjustedHourlyValue == 0.00E0)
MonitorHourlyUnadjustedValueStatus = true

Else
return result A

= All Other Codes:

If (***MatsMhvRecord***.ModcCode in set ***MatsMhvMeasuredModcList***)

If (***MatsMhvRecord***.UnadjustedHourlyValue is null)
return result B

Else if (***MatsMhvRecord***.UnadjustedHourlyValue is not reported in scientific notation to three significant digits)
return result C

Else if (***MatsMhvRecord***.UnadjustedHourlyValue < 0.00E0)
return result D

Else
MonitorHourlyUnadjustedValueStatus = true
MatsMhvCalculatedValue = ***MatsMhvRecord***.UnadjustedHourlyValue

If (***CurrentMhvMaxMinValue*** is not null AND ***MatsMhvRecord***.UnadjustedHourlyValue > ***CurrentMhvMaxMinValue***)
return result E

Else
If (***MatsMhvRecord***.UnadjustedHourlyValue is not null)
return result F

Results:

Result	Response	Severity
A	You reported an MODCCode of 21 in the MATS Monitor Hourly Value record for [param], but the UnadjustedHourlyValue does not equal 0.	Critical Error Level 1
B	You reported a measured value MODCCode in the MATS Monitor Hourly Value record for [param] but did not report an UnadjustedHourlyValue.	Critical Error Level 1
C	The [fieldname] value in the [key] records is not reported in scientific notation rounded to three significant figures, with one digit to the left of the decimal point.	Critical Error Level 1
D	You reported a negative value, which is invalid, in the field [fieldname] for [key].	Critical Error Level 1
E	Warning: The UnadjustedHourlyValue reported in the MATS MHV record for [param] is in excess of the maximum value listed in the monitoring plan. Sources are required to periodically (at least once annually) evaluate the appropriateness of these maximum values in the monitoring plan and make proper adjustments when necessary. Adjustments may include the need to update range values. You should investigate the cause of these exceedances and determine whether adjustments to your monitoring systems or monitoring plan are necessary.	Informational Message
F	You did not report a measured value MODCCode in the MATS Monitor Hourly Value record for [param], but did report an UnadjustedHourlyValue.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-13

Check Name: MATS: QA Status Required and QA Status Parameters

Related Former Checks:

Applicability:

Description: Sets QA Status value and is required parameters.

Specifications:

Set ***QaStatusComponentBeginDate*** = ***MatsMhvRecord***.ComponentBeginDate
Set ***QaStatusComponentId*** = ***MatsMhvRecord***.ComponentId
Set ***QaStatusComponentIdentifier*** = ***MatsMhvRecord***.ComponentIdentifier
Set ***QaStatusComponentTypeCode*** = ***MatsMhvRecord***.ComponentTypeCode
Set ***QaStatusSystemDesignationCode*** = ***MatsMhvRecord***.SystemDesignationCode
Set ***QaStatusSystemId*** = ***MatsMhvRecord***.SystemId
Set ***QaStatusSystemIdentifier*** = ***MatsMhvRecord***.SystemIdentifier
Set ***QaStatusSystemTypeCode*** = ***MatsMhvRecord***.SystemTypeCode

If ***MatsMhvRecord***.ParameterCode is equal to "HGC", "HCLC" or "HFC",

Locate the earliest record in ***EmLocationProgramRecords*** based on EmissionsRecordingBeginDate where:

1) ProgramCode is equal to 'MATS'.
2) EmissionsRecordingBeginDate is less than or equal to the Date of ***CurrentDateHour***.
3) EndDate is null or is greater than or equal to the Date of ***CurrentDateHour***.

if found
Set ***QaStatusMatsErbDate*** to EmissionsRecordingBeginDate in the record located in ***LocationProgramRecords***.
else
Set ***QaStatusMatsErbDate*** to null.
else
Set ***QaStatusMatsErbDate*** to null.

Set ***DailyCalStatusRequired*** = false.
Set ***LinearityStatusRequired*** = false.
Set ***QuarterlyGasAuditStatus*** = false.
Set ***RataStatusRequired*** = false.
Set ***WsiStatusRequired*** = false.

if (***MonitorHourlyModcStatus*** == true) AND (***MatsMhvRecord***.ModcCode in ***MatsMhvMeasuredModcList***) AND (***MatsMhvRecord***.UnadjustedHourlyValue is not null)

if (***MonitorHourlyComponentStatus*** = true) AND (***MatsMhvRecord***.ComponentID is not null)

if (***MatsMhvRecord***.ParameterCode is equal to "HGC")

If (***MatsMhvRecord***.ComponentTypeCode is equal to "HG")

Set ***DailyCalStatusRequired*** = true.
Set ***LinearityStatusRequired*** = true.
Set ***WsiStatusRequired*** = true.

else if (***MatsMhvRecord***.ParameterCode is in set {"HCLC", "HFC"})

Set ***QuarterlyGasAuditStatus*** = true.

if (***MonitorHourlySystemStatus*** = true) AND (***MatsMhvRecord***.MonitoringSystemID is not null)

Set ***RataStatusRequired*** = true.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-14

Check Name: MATS HgC: Complete

Related Former Checks:

Applicability:

Description: Assigns the calculated values for MATS Hg Concentration Monitor Hourly.

Specifications:

MatsMhvCalculatedHgcValue = ***MatsMhvCalculatedValue***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-15

Check Name: MATS HClC: Complete

Related Former Checks:

Applicability:

Description: Assigns the calculated values for MATS HCl Concentration Monitor Hourly.

Specifications:

MatsMhvCalculatedHclcValue = ***MatsMhvCalculatedValue***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-16

Check Name: MATS HFC: Complete

Related Former Checks:

Applicability:

Description: Assigns the calculated values for MATS HF Concentration Monitor Hourly.

Specifications:

MatsMhvCalculatedHfcValue = ***MatsMhvCalculatedValue***

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-17

Check Name: MATS: QA Status Analyzer Range Parameters

Related Former Checks:

Applicability:

Description: Sets analyzer range parameters for Linearity and Daily Calibration QA status checking.

Specifications:

If (***LinearityStatusRequired*** == true) OR (***DailyCalStatusRequired*** == true)

Set ***DualRangeStatus*** = false.

Set ***ApplicableComponentID*** = null.
Set ***ApplicableSystemIDs*** = null.
Set ***CurrentAnalyzerRangeUsed*** = null.
Set ***HighRangeComponentID*** = null.
Set ***LowRangeComponentID*** = null.

Locate a record in ***AnalyzerRangeRecordsByHourLocation*** for the hour and location where the ComponentID is equal to the ***QaStatusComponentId***.

If (***AnalyzerRangeRecordsByHourLocation*** is not found OR if more than one ***AnalyzerRangeRecordsByHourLocation*** is found)

Set ***LinearityStatusRequired*** = false
Set ***DailyCalStatusRequired*** = false

return result A

Else if (***AnalyzerRangeRecordsByHourLocation***.DualRangeIndicator = 1)

Set ***LinearityStatusRequired*** = false
Set ***DailyCalStatusRequired*** = false

return result B

Else if (***AnalyzerRangeRecordsByHourLocation***.AnalyzerRangeCode <> "H")

Set ***LinearityStatusRequired*** = false
Set ***DailyCalStatusRequired*** = false

return result C

Else

Set ***CurrentAnalyzerRangeUsed*** = ***AnalyzerRangeRecordsByHourLocation***.AnalyzerRangeCode.

Set ***ApplicableComponentID*** = ***QaStatusComponentId***.
Set ***HighRangeComponentID*** = ***QaStatusComponentId***.

For each record in ***MonitorSystemComponentRecordsByHourLocation*** where the ComponentID is equal to the ***ApplicableComponentID***

Append ***MonitorSystemComponentRecordsByHourLocation***.SystemID to ***ApplicableSystemIDs***.

if (***MonitorSystemComponentRecordsByHourLocation*** is not found)

set ***LinearityStatusRequired*** = false

set ***DailyCalStatusRequired*** = false

return result D

Results:

Result	Response	Severity
A	You did not report one (and only one) valid Analyzer Range record in your monitoring plan for ComponentID [COMPID] for this hour. The QA Status of the linearity and/or daily calibration tests for this component will not be evaluated.	Critical Error Level 1
B	You reported that ComponentID [COMPID] is a dual range analyzer, but dual range analyzers are not allowed for MATS. The QA Status of the linearity and/or daily calibration tests for this component will not be evaluated.	Critical Error Level 1
C	You reported that ComponentID [COMPID] is not a high range analyzer, but only a high range analyzer is allowed for MATS. The QA Status of the linearity and/or daily calibration tests for this component will not be evaluated.	Critical Error Level 1
D	You did not report any System Component records for ComponentID [compid] in your monitoring plan for the hour. The QA Status of the linearity and/or daily calibration tests for this component will not be evaluated.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS HCl Concentration Monitor Hourly Evaluation
2	Process/Category:	Emissions Data Evaluation Report ------ MATS HF Concentration Monitor Hourly Evaluation
3	Process/Category:	Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-19

Check Name: MATS HgC: 3-Level System Integrity Status Check

Related Former Checks:

Applicability:

Description: Ensures that a 3-Level system integrity was performed for a non-like-kind analyzer component. If a 120 or 125 certification event occurred, this check ensures that either 168 hours or 90/180 has not elapsed since the event, or that a 3-Level system integrity was performed after the event.

Specifications:

Set ***MatsHg3LevelSiTesttRecord*** to null.
Set ***MatsHg3LevelSiEventRecord*** to null.
Set ***MatsHg3LevelSiMissingOpSuppData*** to null.

If ***WsiStatusRequired*** is equal to true, AND ***QaStatusComponentIdentifier*** does not begin with "LK"

Set *CertEventRecord* to null.

Locate the most recent ***Mats3LevelSystemIntegrityRecordsForQaStatus*** record where:

1) ComponentID is equal to ***QaStatusComponentId***.
2) EndDateHour is prior to ***CurrentDateHour***, OR EndDateHour is ***CurrentDateHour***, EndMinute is less than 45.
3) TestResult is equal to "PASSED" or "PASSAPS".

If found,
Set ***MatsHg3LevelSiTesttRecord*** to the located record in ***Mats3LevelSystemIntegrityRecordsForQaStatus***.

If ***MatsHg3LevelSiTesttRecord*** is NOT null,

Locate the most recent record in ***QACertificationEventRecords*** where:

1a) QaCertEventDateHour is equal to ***CurrentDateHour*** and ConditionalBeginDateHour, OR
1b) QaCertEventDateHour is prior to ***CurrentDateHour***.
2) ComponentID is equal to ***QaStatusComponentId***.
3) QaCertEventCode is equal to "100", "101", "120" or "125".
4) QaCertEventDateHour is after ***MatsHg3LevelSiTesttRecord***.EndDateHour.

If found,
Set *CertEventRecord* to the located record in ***QACertificationEventRecords***

Else
Locate the most recent record in ***QACertificationEventRecords*** where:

1a) QaCertEventDateHour is equal to ***CurrentDateHour*** and ConditionalBeginDateHour, OR
1b) QaCertEventDateHour is prior to ***CurrentDateHour***.
2) ComponentID is equal to ***QaStatusComponentId***.
3) QaCertEventCode is equal to "100", "101", "120" or "125".

If found,
Set *CertEventRecord* to the located record in ***QACertificationEventRecords***

If *CertEventRecord* is NOT null,

Set ***MatsHg3LevelSiEventRecord*** to *CertEventRecord*.

If *CertEventRecord*.ConditionalBeginDateHour is NOT null, AND is on or prior to ***CurrentDateHour***,

Determine *ConditionalDataStatus*:

When *CertEventRecord*.QaCertEventCode is equal to “125”:

1) If ***QaStatusComponentBeginDate*** is null, set *ConditionalDataStatus* to **EXPIRED**.
2) Locate a record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where ProgramCode equals “MATS” and UnitMonitorCertBeginDate is on or before ***QaStatusComponentBeginDate***.
3) If not found, locate a record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where ProgramCode equals “MATS” and EmissionsRecordingBeginDate on or before ***QaStatusComponentBeginDate***.
4) If a ***LocationProgramRecordsByHourLocation*** was not located, set *ConditionalDataStatus* to **MISSINGPROGRAM**.
5) Else if UnitMonitorCertDeadline of the located record is NOT null, AND is on or prior to the date of ***CurrentDateHour***, set *ConditionalDataStatus* to **EXPIRED**.
6) Else if UnitMonitorCertDeadline of the located record is null, AND UnitMonitorCertBeginDate + 180 is on or prior to the date of ***CurrentDateHour***, set *ConditionalDataStatus* to **EXPIRED**.
7) Otherwise set *ConditionalDataStatus* to **VALID**.

When *CertEventRecord*.QaCertEventCode is equal to "100", "101" or “120”:

1) If *CertEventRecord*.ConditionalBeginDateHour is null, set *ConditionalDataStatus* to **EXPIRED**.
2) Else if the number of clock hours on or after *CertEventRecord*.ConditionalBeginDateHour and on or before ***CurrentDateHour*** is less than or equal to 168, set *ConditionalDataStatus* to **VALID**.
3) Else if *CertEventRecord*.ConditionalBeginDateHour and ***CurrentDateHour*** are in the same quarter,
   a) Count the ***HourlyOperatingDataRecordsForLocation*** where:
      • OpTime is greater than 0.
      • DateHour is on or after *CertEventRecord*.ConditionalBeginDateHour.
      • DateHour is on or before ***CurrentDateHour***.
   b) If count is greater than 168, set *ConditionalDataStatus* to **EXPIRED**.
   c) Otherwise set *ConditionalDataStatus* to **VALID**.
4) Else
   /* Grab the operating hours for the current quarter on or before the current hour */
   a) Set *OperatingHoursCurrentQuarter* to:
      • The value of ***RptPeriodOpHoursAccumulatorArray*** for the location when it is not -1.
      • Otherwise, 0.
   b) If *OperatingHoursCurrentQuarter* is greater than 168, set *ConditionalDataStatus* to **EXPIRED**.
   /* Find sum of Op Hours for supplemental record between the quarter of the event quarter and the current quarter */
   c) Else set *OperatingHoursBetweenQuarters* to the sum of OpValue for ***OperatingSuppDataRecordsByLocation*** where:
      • OpTypeCode equals “OPHOURS”.
      • FuelCode is null.
      • ReportingPeriod is for a quarter after the quarter of *CertEventRecord*.ConditionalBeginDateHour and before the quarter of ***CurrentDateHour***.
   /* Determine whether the operating hours for the current and ‘between’ quarters exceed the allowed */
   d) If *OperatingHoursCurrentQuarter* + *OperatingHoursBetweenQuarters* is greater than 168, set *ConditionalDataStatus* to **EXPIRED**.

/* Stop checking if subsequent checks are affected by missing data */
e) Else if value of ***RptPeriodOpHoursAccumulatorArray*** for the location is -1, set *ConditionalDataStatus* to **MISSINGACCUM**.
f) Else if an ***OperatingSuppDataRecordsByLocation*** was missing for any quarter, set *ConditionalDataStatus* to **MISSINGOPSUPP** and append the quarter description to ***MatsHg3LevelSiMissingOpSuppData***.
/* Use QA Cert Event Supplemental Data for Conditional Begin Date if it exists */
g) Else if *CertEventRecord*.ConditionalBeginHourSuppDataExists is true.
    i) If *OperatingHoursCurrentQuarter* + *OperatingHoursBetweenQuarters* + *CertEventRecord*.ConditionalBeginOpHoursCount is less than or equal to 168, set *ConditionalDataStatus* to **VALID**.
    ii) Otherwise set *ConditionalDataStatus* to **EXPIRED**.
/* Find Op Hours supplemental record for the quarter of the event */
h) Else set *OperatingHoursEventQuarter* to OpValue of the ***OperatingSuppDataRecordsByLocation*** where:
    • OpTypeCode equals “OPHOURS”.
    • FuelCode is null.
    • ReportingPeriod is the quarter of *CertEventRecord*.ConditionalBeginDateHour.
i) If a record was not found, set *ConditionalDataStatus* to **MISSINGOPSUPP** and append the quarter description to ***MatsHg3LevelSiMissingOpSuppData***.
/* Check whether assuming that every hour in the event quarter is operating would not exceed allowed */
j) Else if *OperatingHoursCurrentQuarter* + *OperatingHoursBetweenQuarters* + *OperatingHoursEventQuarter* is less than or equal to 168, set *ConditionalDataStatus* to **VALID**.
/* Check whether assuming the minimum number of operating hours in the event quarter would exceeding allowed */
k) Else if *OperatingHoursEventQuarter* is greater than the number of clock hours in the quarter prior to *CertEventRecord*.ConditionalBeginDateHour,
l) And if *OperatingHoursCurrentQuarter* + *OperatingHoursBetweenQuarters* + (*OperatingHoursEventQuarter* minus the number of prior clock hours) is greater than 168,
m) Then set *ConditionalDataStatus* to **EXPIRED**
/* Check that treating every calendar hour on or after the conditional data begin hour as an operating hour does not exceed allowed */
n) Else if *OperatingHoursEventQuarter* is greater than the number of clock hours in the quarter on or after *CertEventRecord*.ConditionalBeginDateHour,
o) And if *OperatingHoursCurrentQuarter* + *OperatingHoursBetweenQuarters* + the number of on or after clock hours is less than or equal to 168
p) Then set *ConditionalDataStatus* to **VALID**,
/* Cannot determine whether allowed operating hours were exceeded because of uncertainty about operating hours in event quarter */
q) Else set *ConditionalDataStatus* to **UNDETERMINED**

If *ConditionalDataStatus* is equal to **EXPIRED**,

    If *SystemIntegrityRecord* is null,

        return result A.

    Else

        return result B.

Else if *ConditionalDataStatus* is equal to **UNDETERMINED**,

    If *SystemIntegrityRecord* is null,

return result D.

Else

return result E.

Else if *ConditionalDataStatus* is equal to **MISSINGPROGRAM**,

return result F.

Else if *ConditionalDataStatus* is equal to **MISSINGACCUM**,

return result G.

Else if *ConditionalDataStatus* is equal to **MISSINGOPSUPP**,

return result H.

Else if *ConditionalDataStatus* is equal to **MISSINGVALUE**,

return result I.

Else

return result J.

Else if ***MatsHg3LevelSiTesttRecord*** is null

return result C.

Else if ***MatsHg3LevelSiTesttRecord***.QaNeedsEvaluationFlag is equal to "Y",

return result K.

Results:

Result	Response	Severity
A	The conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [key] has expired.	Critical Error Level 1
B	The conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [key] has expired. A prior test was ignored.	Critical Error Level 1
C	You did not report a prior [testtype] or certification event for [key].	Critical Error Level 1
D	The software could not determine if the current hour was within the conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [key]	Informational Message
E	The software could not determine if the current hour was within the conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [key]	Informational Message
F	The [testtype] status for [key] could not be determined, because a Unit Program record associated with the initial certification event for QACertEventCode [code] QACertEventDate [eventdate] either does not exist or has a UnitMonitorCertificationBeginDate inconsistent with the BeginDate of the associated Monitor System record.	Critical Error Level 1
G	The [testtype] status for [key] could not be determined, because the OperatingTime in at least one Hourly Operating Data records was missing or invalid.	Critical Error Level 1
H	The [testtype] status for [key] could not be determined, because the Op Supp Data record for OPHOURS, OSHOURS, or OPDAYS is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
I	A prior required parameter for check execution has failed to load. Please contact technical support.	Critical Error Level 1
J	You reported a QA Certification Event record for Component [key], QACertEventCode [code] and QACertEventDate [eventdate], but the conditional data period has not started.	Critical Error Level 1
K	The [testtype] status for [key] could not be determined, because the applicable prior [testtype] with TestNumber [testnum] has not yet been evaluated.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Code: MATSMHV-20

Check Name: MATS HgC: Linearity Status Check

Related Former Checks:

Applicability:

Description: Ensures that a Hg Linearity was performed for a non-like-kind analyzer component. If a 120 or 125 certification event occurred, this check ensures that either 168 hours or 90/180 has not elapsed since the event, or that an Hg Linearity was performed after the event.

Specifications:

Set ***MatsHgLinearityTestRecord*** to null.
Set ***MatsHgLinearityEventRecord*** to null.
Set ***MatsHgLinearityMissingOpSuppData*** to null.

If ***WsiStatusRequired*** is equal to true, AND ***QaStatusComponentIdentifier*** does not begin with "LK"

Set *CertEventRecord* to null.

Locate the most recent ***Mats3LevelSystemIntegrityRecordsForQaStatus*** record where:

1) ComponentID is equal to ***QaStatusComponentId***.
2) EndDateHour is prior to ***CurrentDateHour***, OR EndDateHour is ***CurrentDateHour***, EndMinute is less than 45,
3) TestResult is equal to "PASSED" or "PASSAPS".

If found,

Set ***MatsHgLinearityTestRecord*** to the located record in ***Mats3LevelSystemIntegrityRecordsForQaStatus***.

If ***MatsHgLinearityTestRecord*** is NOT null,

Locate the most recent record in ***QACertificationEventRecords*** where:

1a) QaCertEventDateHour is equal to ***CurrentDateHour*** and ConditionalBeginDateHour, OR
1b) QaCertEventDateHour is prior to ***CurrentDateHour***.
2) ComponentID is equal to ***QaStatusComponentId***.
3) QaCertEventCode is equal to "100", "101", "120" or "125".
4) QaCertEventDateHour is after ***MatsHgLinearityTestRecord***.EndDateHour.

If found,
Set *CertEventRecord* to the located record in ***QACertificationEventRecords***

Else

Locate the most recent record in ***QACertificationEventRecords*** where:

1a) QaCertEventDateHour is equal to ***CurrentDateHour*** and ConditionalBeginDateHour, OR
1b) QaCertEventDateHour is prior to ***CurrentDateHour***.
2) ComponentID is equal to ***QaStatusComponentId***.
3) QaCertEventCode is equal to "100", "101", "120" or "125".

If found,
Set *CertEventRecord* to the located record in ***QACertificationEventRecords***

If *CertEventRecord* is NOT null

Set ***MatsHgLinearityEventRecord*** to *CertEventRecord*.

If *CertEventRecord*.ConditionalBeginDateHour is NOT null, AND is on or prior to ***CurrentDateHour***,

Determine *ConditionalDataStatus*:

When *CertEventRecord*.QaCertEventCode is equal to “125”:

1) If ***QaStatusComponentBeginDate*** is null, set *ConditionalDataStatus* to **EXPIRED**.
2) Locate a record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where ProgramCode equals “MATS” and UnitMonitorCertBeginDate is on or before ***QaStatusComponentBeginDate***.
3) If not found, locate a record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where ProgramCode equals “MATS” and EmissionsRecordingBeginDate on or before ***QaStatusComponentBeginDate***.
4) If a ***LocationProgramRecordsByHourLocation*** was not located, set *ConditionalDataStatus* to **MISSINGPROGRAM**.
5) Else if UnitMonitorCertDeadline of the located record is NOT null, AND is on or prior to the date of ***CurrentDateHour***, set *ConditionalDataStatus* to **EXPIRED**.
6) Else if UnitMonitorCertDeadline of the located record is null, AND UnitMonitorCertBeginDate + 180 is on or prior to the date of ***CurrentDateHour***, set *ConditionalDataStatus* to **EXPIRED**.
7) Otherwise set *ConditionalDataStatus* to **VALID**.

When *CertEventRecord*.QaCertEventCode is equal to "100", "101" or "120":

1) If *CertEventRecord*.ConditionalBeginDateHour is null, set *ConditionalDataStatus* to **EXPIRED**.
2) Else if the number of clock hours on or after *CertEventRecord*.ConditionalBeginDateHour and on or before ***CurrentDateHour*** is less than or equal to 168, set *ConditionalDataStatus* to **VALID**.
3) Else if *CertEventRecord*.ConditionalBeginDateHour and ***CurrentDateHour*** are in the same quarter,
   a) Count the ***HourlyOperatingDataRecordsForLocation*** where:
      • OpTime is greater than 0.
      • DateHour is on or after *CertEventRecord*.ConditionalBeginDateHour.
      • DateHour is on or before ***CurrentDateHour***.
   b) If count is greater than 168, set *ConditionalDataStatus* to **EXPIRED**.
   c) Otherwise set *ConditionalDataStatus* to **VALID**.
4) Else
   /* Grab the operating hours for the current quarter on or before the current hour */
   a) Set *OperatingHoursCurrentQuarter* to:
      • The value of ***RptPeriodOpHoursAccumulatorArray*** for the location when it is not -1.
      • Otherwise, 0.
   b) If *OperatingHoursCurrentQuarter* is greater than 168, set *ConditionalDataStatus* to **EXPIRED**.
   /* Find sum of Op Hours for supplemental record between the quarter of the event quarter and the current quarter */
   c) Else set *OperatingHoursBetweenQuarters* to the sum of OpValue for ***OperatingSuppDataRecordsByLocation*** where:
      • OpTypeCode equals “OPHOURS”.
      • FuelCode is null.
      • ReportingPeriod is for a quarter after the quarter of *CertEventRecord*.ConditionalBeginDateHour and before the quarter of ***CurrentDateHour***.
   /* Determine whether the operating hours for the current and ‘between’ quarters exceed the allowed */
   d) If *OperatingHoursCurrentQuarter* + *OperatingHoursBetweenQuarters* is greater than

168, set *ConditionalDataStatus* to **EXPIRED**.
/* Stop checking if subsequent checks are affected by missing data */
e) Else if value of ***RptPeriodOpHoursAccumulatorArray*** for the location is -1, set *ConditionalDataStatus* to **MISSINGACCUM**.
f) Else if an ***OperatingSuppDataRecordsByLocation*** was missing for any quarter, set *ConditionalDataStatus* to **MISSINGOPSUPP** and append the quarter description to ***MatsHgLinearityMissingOpSuppData***.
/* Use QA Cert Event Supplemental Data for Conditional Begin Date if it exists */
g) Else if *CertEventRecord*.ConditionalBeginHourSuppDataExists is true.
i) If *OperatingHoursCurrentQuarter* + *OperatingHoursBetweenQuarters* + *CertEventRecord*.ConditionalBeginOpHoursCount is less than or equal to 168, set *ConditionalDataStatus* to **VALID**.
ii) Otherwise set *ConditionalDataStatus* to **EXPIRED**.
/* Find Op Hours supplemental record for the quarter of the event */
h) Else set *OperatingHoursEventQuarter* to OpValue of the ***OperatingSuppDataRecordsByLocation*** where:
• OpTypeCode equals "OPHOURS".
• FuelCode is null.
• ReportingPeriod is the quarter of *CertEventRecord*.ConditionalBeginDateHour.
i) If a record was not found, set *ConditionalDataStatus* to **MISSINGOPSUPP** and append the quarter description to ***MatsHgLinearityMissingOpSuppData***.
/* Check whether assuming that every hour in the event quarter is operating would not exceed allowed */
j) Else if *OperatingHoursCurrentQuarter* + *OperatingHoursBetweenQuarters* + *OperatingHoursEventQuarter* is less than or equal to 168, set *ConditionalDataStatus* to **VALID**.
/* Check whether assuming the minimum number of operating hours in the event quarter would exceeding allowed */
k) Else if *OperatingHoursEventQuarter* is greater than the number of clock hours in the quarter prior to *CertEventRecord*.ConditionalBeginDateHour,
l) And if *OperatingHoursCurrentQuarter* + *OperatingHoursBetweenQuarters* + (*OperatingHoursEventQuarter* minus the number of prior clock hours) is greater than 168,
m) Then set *ConditionalDataStatus* to **EXPIRED**
/* Check that treating every calendar hour on or after the conditional data begin hour as an operating hour does not exceed allowed */
n) Else if *OperatingHoursEventQuarter* is greater than the number of clock hours in the quarter on or after *CertEventRecord*.ConditionalBeginDateHour,
o) And if *OperatingHoursCurrentQuarter* + *OperatingHoursBetweenQuarters* + the number of on or after clock hours is less than or equal to 168
p) Then set *ConditionalDataStatus* to **VALID**,
/* Cannot determine whether allowed operating hours were exceeded because of uncertainty about operating hours in event quarter */
q) Else set *ConditionalDataStatus* to **UNDETERMINED**

If *ConditionalDataStatus* is equal to **EXPIRED**,

If *HgLinearityRecord* is null,

return result A.

Else

return result B.

Else if *ConditionalDataStatus* is equal to **UNDETERMINED**,

If *HgLinearityRecord* is null,

return result D.

Else

return result E.

Else if *ConditionalDataStatus* is equal to **MISSINGPROGRAM**,

return result F.

Else if *ConditionalDataStatus* is equal to **MISSINGACCUM**,

return result G.

Else if *ConditionalDataStatus* is equal to **MISSINGOPSUPP**,

return result H.

Else if *ConditionalDataStatus* is equal to **MISSINGVALUE**,

return result I.

Else

return result J.

Else if ***MatsHgLinearityTestRecord*** is null

return result C.

Else if ***MatsHgLinearityTestRecord***.QaNeedsEvaluationFlag is equal to "Y",

return result K.

Results:

Result	Response	Severity
A	The conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [key] has expired.	Critical Error Level 1
B	The conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [key] has expired. A prior test was ignored.	Critical Error Level 1
C	You did not report a prior [testtype] or certification event for [key].	Critical Error Level 1
D	The software could not determine if the current hour was within the conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [key]	Informational Message
E	The software could not determine if the current hour was within the conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for [key]	Informational Message
F	The [testtype] status for [key] could not be determined, because a Unit Program record associated with the initial certification event for QACertEventCode [code] QACertEventDate [eventdate] either does not exist or has a UnitMonitorCertificationBeginDate inconsistent with the BeginDate of the associated Monitor System record.	Critical Error Level 1
G	The [testtype] status for [key] could not be determined, because the OperatingTime in at least one Hourly Operating Data records was missing or invalid.	Critical Error Level 1
H	The [testtype] status for [key] could not be determined, because the Op Supp Data record for OPHOURS, OSHOURS, or OPDAYS is missing for one or more previous reporting periods. If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
I	A prior required parameter for check execution has failed to load. Please contact technical support.	Critical Error Level 1
J	You reported a QA Certification Event record for Component [key], QACertEventCode [code] and QACertEventDate [eventdate], but the conditional data period has not started.	Critical Error Level 1
K	The [testtype] status for [key] could not be determined, because the applicable prior [testtype] with TestNumber [testnum] has not yet been evaluated.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ MATS Hg Concentration Monitor Hourly Evaluation

Check Category:

MATS Operating Hour Checks

Check Code: MATSHOD-1

Check Name: MATS Hg: Locate Active Monitor Method

Related Former Checks:

Applicability:

Description: Locates the active Monitor Method record with Parameter Code equal to 'HGRE' or 'HGRH' and for the current hour and location. Returns a negative result if more than one active record is found. Returns the record and the Parameter Code in the record as output parameters if one record is found.

Specifications:

Set ***MatsHgMethodRecord*** to null.
Set ***MatsHgParameterCode*** to null.
Set ***MatsHgMethodCode*** to null.

If ***DerivedHourlyChecksNeeded*** is equal to true,

Locate ***MonitorMethodRecordsByHourLocation*** records where ParameterCode is equal to "HGRE" or "HGRH".

If more than one record was located,

return result A

Else if one record was located,

Set ***MatsHgMethodRecord*** to the located record.
Set ***MatsHgParameterCode*** to ***MatsHgMethodRecord***.ParameterCode.
Set ***MatsHgMethodCode*** to ***MatsHgMethodRecord***.MethodCode.

If ***MatsHgMethodCode*** is equal to "ST" or "CEMST",,

Set ***FlowMhvOptionallyAllowed*** to true.

Results:

Result	Response	Severity
A	You reported more than one monitoring method record for [param] for the hour and location.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-2

Check Name: MATS HCl: Locate Active Monitor Method

Related Former Checks:

Applicability:

Description: Locates the active Monitor Method record with Parameter Code equal to 'HCLRE' or 'HCLRH' and for the current hour and location. Returns a negative result if more than one active record is found. Returns the record and the Parameter Code in the record as output parameters if one record is found.

Specifications:

Set ***MatsHclMethodRecord*** to null.
Set ***MatsHclParameterCode*** to null.
Set ***MatsHclMethodCode*** to null.

If ***DerivedHourlyChecksNeeded*** is equal to true,

Locate ***MonitorMethodRecordsByHourLocation*** records where ParameterCode is equal to "HCLRE" or "HCLRH".

If more than one record was located,

return result A

Else if one record was located,

Set ***MatsHclMethodRecord*** to the located record.
Set ***MatsHclParameterCode*** to ***MatsHclMethodRecord***.ParameterCode.
Set ***MatsHclMethodCode*** to ***MatsHclMethodRecord***.MethodCode.

Results:

Result	Response	Severity
A	You reported more than one monitoring method record for [param] for the hour and location.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-3

Check Name: MATS HF: Locate Active Monitor Method

Related Former Checks:

Applicability:

Description: Locates the active Monitor Method record with Parameter Code equal to 'HFRE' or 'HFRH' and for the current hour and location. Returns a negative result if more than one active record is found. Returns the record and the Parameter Code in the record as output parameters if one record is found.

Specifications:

Set ***MatsHfMethodRecord*** to null.
Set ***MatsHfParameterCode*** to null.
Set ***MatsHfMethodCode*** to null.

If ***DerivedHourlyChecksNeeded*** is equal to true,

Locate ***MonitorMethodRecordsByHourLocation*** records where ParameterCode is equal to "HFRE" or "HFRH".

If more than one record was located,

return result A

Else if one record was located,

Set ***MatsHfMethodRecord*** to the located record.
Set ***MatsHfParameterCode*** to ***MatsHfMethodRecord***.ParameterCode.
Set ***MatsHfMethodCode*** to ***MatsHfMethodRecord***.MethodCode.

Results:

Result	Response	Severity
A	You reported more than one monitoring method record for [param] for the hour and location.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-4

Check Name: MATS SO2: Locate Active Monitor Method

Related Former Checks:

Applicability:

Description: Locates the active Monitor Method record with Parameter Code equal to 'SO2RE' or 'SO2RH' and for the current hour and location. Returns a negative result if more than one active record is found. Returns the record and the Parameter Code in the record as output parameters if one record is found.

Specifications:

Set ***MatsSo2MethodRecord*** to null.
Set ***MatsSo2ParameterCode*** to null.
Set ***MatsSo2MethodCode*** to null.

If ***DerivedHourlyChecksNeeded*** is equal to true,

Locate ***MonitorMethodRecordsByHourLocation*** records where ParameterCode is equal to "SO2RE" or "SO2RH".

If more than one record was located,

return result A

Else if one record was located,

Set ***MatsSo2MethodRecord*** to the located record.
Set ***MatsSo2ParameterCode*** to ***MatsSo2MethodRecord***.ParameterCode.
Set ***MatsSo2MethodCode*** to ***MatsSo2MethodRecord***.MethodCode.

Results:

Result	Response	Severity
A	You reported more than one monitoring method record for [param] for the hour and location.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-5

Check Name: MATS: Set MATS Expected Flag

Related Former Checks:

Applicability:

Description: Uses whether MATS Hg Method Code, MATS HCl Method Code, MATS HF Method Code or MATS SO2 Method Code is not null to determine whether MATS is expected.

Specifications:

If ***MatsHgParameterCode***, ***MatsHclParameterCode***, ***MatsHfParameterCode*** or ***MatsSo2ParameterCode*** is not null,

Set ***MatsExpected*** to true.

Else

Set ***MatsExpected*** to false.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-6

Check Name: MATS Hg: Locate Derived Hourly Record

Related Former Checks:

Applicability:

Description: Locates the MATS Derived Hourly record with Parameter Code equal to 'HGRE' or 'HGRH' and for the current hour and location. Returns a negative result if more than one active record is found. Returns the record, flags indicating whether 'HGRE' or 'HGRH' checks are needed, and a flag indicating that Hg Concentration is needed, if one record is found.

Specifications:

Set ***MatsHgreDhvChecksNeeded*** to false.
Set ***MatsHgrhDhvChecksNeeded*** to false.
Set ***MatsHgcNeeded*** to false.
Set ***MatsHgDhvRecord*** to null.
Set ***MatsHgDhvParameterDescription*** to "MATS Hg Rate".

If ***DerivedHourlyChecksNeeded*** is equal to true,

Set *RecordCount* equal to the number of records in ***MatsDhvRecordsByHourLocation*** where ParameterCode is equal to "HGRE" or "HGRH".

If ***CurrentHourlyOpRecord***.OperatingTime is greater than 0

If ***MatsHgParameterCode*** is null,

If *RecordCount* is greater than 0,

return result A.

Else /* Method Exists for Hg */

If *RecordCount* is equal to 0,

return result E

Else if *RecordCount* is greater than 1,

return result B.

Else /* *RecordCount* is equal to 1 */

Set ***MatsHgDhvRecord*** to the located ***MatsDhvRecordsByHourLocation*** record
Append ***MatsHgDhvRecord***.UnadjustedHourlyValue to ***ApportionmentHgRateArray***
Append ***MatsHgDhvRecord***.ModcCode to ***MatsMS1HgModcCodeArray***

If ***MatsHgDhvRecord***.ParameterCode is equal to ***MatsHgParameterCode***

If ***MatsHgMethodCode*** is equal to "CALC",

If ***MatsHgDhvRecord***.EquationCode is null,

If ***MatsHgDhvRecord***.ModcCode is NOT equal to "38",
return result F.
Else
return result H.

Else

If ***MatsHgDhvRecord***.EquationCode is NOT equal to "MS-1",
return result F.
Else if ***MatsHgDhvRecord***.ModcCode is NOT equal to "38",
Set ***MatsMs1HgDhvId*** to ***MatsHgDhvRecord***.MatsDhvId
Set ***MatsMs1HgUnadjustedHrlyValue*** to
MatsHgDhvRecord.UnadjustedHrlyValue
Set ***MatsParameterPluginHg*** to ***MatsHgDhvRecord***.ParameterCode

Else

If ***MatsHgDhvRecord***.EquationCode is equal to "MS-1",

return result G.

Else

Set ***MatsHgcNeeded*** to true.

If ***MatsHgDhvRecord***.ParameterCode is equal to 'HGRE',

Set ***MatsHgreDhvChecksNeeded*** to true.

Else if ***MatsHgDhvRecord***.ParameterCode is equal to 'HGRH',

Set ***MatsHgrhDhvChecksNeeded*** to true.

Else /* DHV and Method parameter code mismatch */

return result C.

Else /* Non Operating Hour */

If *RecordCount* is greater than 0,

return result D.

Results:

Result	Response	Severity
A	MATS Derived Hourly Value records were reported for [param], but no supporting method exists.	Critical Error Level 1
B	You reported more than one MATS Derived Hourly Value records for [param] for the hour.	Critical Error Level 1
C	MATS Derived Hourly Value records were reported for [param], but no supporting method exists.	Critical Error Level 1
D	You reported MATS Derived Hourly Value records for [param] that are not reported if the unit did not operate in the hour.	Critical Error Level 1
E	No required MATS Derived Hourly Value records were reported for [param].	Critical Error Level 1
F	The reported MATS method code is equal to "CALC." However, the appropriate Formula Code "MS-1" has not been reported.	Critical Error Level 1
G	The reported Formula Code of "MS-1" is inconsistent with the reported MATS Method Code.	Critical Error Level 1
H	The reported MATS method code is equal to "CALC." However, the appropriate Formula Code "MS-1" has not been reported.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-7

Check Name: MATS HCl: Locate Derived Hourly Record

Related Former Checks:

Applicability:

Description: Locates the MATS Derived Hourly record with Parameter Code equal to 'HCLRE' or 'HCLRH' and for the current hour and location. Returns a negative result if more than one active record is found. Returns the record, flags indicating whether 'HCLRE' or 'HCLRH' checks are needed, and a flag indicating that Hg Concentration is needed, if one record is found.

Specifications:

Set ***MatsHclreDhvChecksNeeded*** to false.
Set ***MatsHclrhDhvChecksNeeded*** to false.
Set ***MatsHclcNeeded*** to false.
Set ***MatsHclDhvRecord*** to null.
Set ***MatsHclDhvParameterDescription*** to "MATS HCl Rate".

If ***DerivedHourlyChecksNeeded*** is equal to true,

Set *RecordCount* equal to the number of records in ***MatsDhvRecordsByHourLocation*** where ParameterCode is equal to "HCLRE" or "HCLRH".

If ***CurrentHourlyOpRecord***.OperatingTime is greater than 0

If ***MatsHclParameterCode*** is null,

If *RecordCount* is greater than 0,

return result A.

Else /* Method Exists for HCl */

If *RecordCount* is equal to 0,

return result E

Else if *RecordCount* is greater than 1,

return result B.

Else /* *RecordCount* is equal to 1 */

Set ***MatsHclDhvRecord*** to the located ***MatsDhvRecordsByHourLocation*** record.
Append ***MatsHclDhvRecord***.UnadjustedHourlyValue to ***ApportionmentHclRateArray***
Append ***MatsHclDhvRecord***.ModcCode to ***MatsMS1HclModcCodeArray***

If ***MatsHclDhvRecord***.ParameterCode is equal to ***MatsHclParameterCode***

If ***MatsHclMethodCode*** is equal to "CALC",

If ***MatsHclDhvRecord***.EquationCode is null,

If ***MatsHgDhvRecord***.ModcCode is NOT equal to "38",
return result F.
Else
return result H.

Else

If ***MatsHclDhvRecord***.EquationCode is NOT equal to "MS-1",
return result F.
Else if ***MatsHclDhvRecord***.ModcCode is NOT equal to "38",
Set ***MatsMs1HclDhvId*** to ***MatsHclDhvRecord***.MatsDhvId
Set ***MatsMs1HclUnadjustedHrlyValue*** to
MatsHclDhvRecord.UnadjustedHrlyValue
Set ***MatsParameterPluginHcl*** to ***MatsHclDhvRecord***.ParameterCode

Else

If ***MatsHclDhvRecord***.EquationCode is equal to "MS-1",

return result G.

Else

Set ***MatsHclcNeeded*** to true.

If ***MatsHclDhvRecord***.ParameterCode is equal to 'HCLRE',

Set ***MatsHclreDhvChecksNeeded*** to true.

Else if ***MatsHclDhvRecord***.ParameterCode is equal to 'HCLRH',

Set ***MatsHclrhDhvChecksNeeded*** to true.

Else /* DHV and Method parameter code mismatch */

return result C.

Else /* Non Operating Hour */

If *RecordCount* is greater than 0,

return result D.

Results:

Result	Response	Severity
A	MATS Derived Hourly Value records were reported for [param], but no supporting method exists.	Critical Error Level 1
B	You reported more than one MATS Derived Hourly Value records for [param] for the hour.	Critical Error Level 1
C	MATS Derived Hourly Value records were reported for [param], but no supporting method exists.	Critical Error Level 1
D	You reported MATS Derived Hourly Value records for [param] that are not reported if the unit did not operate in the hour.	Critical Error Level 1
E	No required MATS Derived Hourly Value records were reported for [param].	Critical Error Level 1
F	The reported MATS method code is equal to "CALC." However, the appropriate Formula Code "MS-1" has not been reported.	Critical Error Level 1
G	The reported Formula Code of "MS-1" is inconsistent with the reported MATS Method Code.	Critical Error Level 1
H	The reported MATS method code is equal to "CALC." However, the appropriate Formula Code "MS-1" has not been reported.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-8

Check Name: MATS HF: Locate Derived Hourly Record

Related Former Checks:

Applicability:

Description: Locates the MATS Derived Hourly record with Parameter Code equal to 'HFRE' or 'HFRH' and for the current hour and location. Returns a negative result if more than one active record is found. Returns the record, flags indicating whether 'HFRE' or 'HFRH' checks are needed, and a flag indicating that Hg Concentration is needed, if one record is found.

Specifications:

Set ***MatsHfreDhvChecksNeeded*** to false.
Set ***MatsHfrhDhvChecksNeeded*** to false.
Set ***MatsHfcNeeded*** to false.
Set ***MatsHfDhvRecord*** to null.
Set ***MatsHfDhvParameterDescription*** to "MATS HF Rate".

If ***DerivedHourlyChecksNeeded*** is equal to true,

Set *RecordCount* equal to the number of records in ***MatsDhvRecordsByHourLocation*** where ParameterCode is equal to "HFRE" or "HFRH".

If ***CurrentHourlyOpRecord***.OperatingTime is greater than 0

If ***MatsHfParameterCode*** is null,

If *RecordCount* is greater than 0,

return result A.

Else /* Method Exists for HF */

If *RecordCount* is equal to 0,

return result E

Else if *RecordCount* is greater than 1,

return result B.

Else /* *RecordCount* is equal to 1 */

Set ***MatsHfDhvRecord*** to the located ***MatsDhvRecordsByHourLocation*** record.
Append ***MatsHfDhvRecord***.UnadjustedHourlyValue to ***ApportionmentHfRateArray***
Append ***MatsHfDhvRecord***.ModcCode to ***MatsMS1HfModcCodeArray***

If ***MatsHfDhvRecord***.ParameterCode is equal to ***MatsHfParameterCode***

If ***MatsHfMethodCode*** is equal to "CALC",

If ***MatsHfDhvRecord***.EquationCode is null,

If ***MatsHgDhvRecord***.ModcCode is NOT equal to "38",
return result F.
Else
return result H.

Else

If ***MatsHfDhvRecord***.EquationCode is NOT equal to "MS-1",
return result F.
Else if ***MatsHfDhvRecord***.ModcCode is NOT equal to "38",
Set ***MatsMs1HfDhvId*** to ***MatsHfDhvRecord***.MatsDhvId
Set ***MatsMs1HfUnadjustedHrlyValue*** to
MatsHfDhvRecord.UnadjustedHrlyValue
Set ***MatsParameterPluginHf t***o ***MatsHfDhvRecord***.ParameterCode

Else

If ***MatsHfDhvRecord***.EquationCode is equal to "MS-1",

return result G.

Else

Set ***MatsHfcNeeded*** to true.

If ***MatsHfDhvRecord***.ParameterCode is equal to 'HFRE',

Set ***MatsHfreDhvChecksNeeded*** to true.

Else if ***MatsHfDhvRecord***.ParameterCode is equal to 'HFRH',

Set ***MatsHfrhDhvChecksNeeded*** to true.

Else /* DHV and Method parameter code mismatch */

return result C.

Else /* Non Operating Hour */

If *RecordCount* is greater than 0,

return result D.

Results:

Result	Response	Severity
A	MATS Derived Hourly Value records were reported for [param], but no supporting method exists.	Critical Error Level 1
B	You reported more than one MATS Derived Hourly Value records for [param] for the hour.	Critical Error Level 1
C	MATS Derived Hourly Value records were reported for [param], but no supporting method exists.	Critical Error Level 1
D	You reported MATS Derived Hourly Value records for [param] that are not reported if the unit did not operate in the hour.	Critical Error Level 1
E	No required MATS Derived Hourly Value records were reported for [param].	Critical Error Level 1
F	The reported MATS method code is equal to "CALC." However, the appropriate Formula Code "MS-1" has not been reported.	Critical Error Level 1
G	The reported Formula Code of "MS-1" is inconsistent with the reported MATS Method Code.	Critical Error Level 1
H	The reported MATS method code is equal to "CALC." However, the appropriate Formula Code "MS-1" has not been reported.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-9

Check Name: MATS SO2: Locate Derived Hourly Record

Related Former Checks:

Applicability:

Description: Locates the MATS Derived Hourly record with Parameter Code equal to 'SO2RE' or 'SO2RH' and for the current hour and location. Returns a negative result if more than one active record is found. Returns the record, flags indicating whether 'SO2RE' or 'SO2RH' checks are needed, and a flag indicating that Hg Concentration is needed, if one record is found.

Specifications:

Set ***MatsSo2reDhvChecksNeeded*** to false.
Set ***MatsSo2rhDhvChecksNeeded*** to false.
Set ***MatsSo2cNeeded*** to false.
Set ***MatsSo2DhvRecord*** to null.
Set ***MatsSo2DhvParameterDescription*** to "MATS SO2 Rate".

If ***DerivedHourlyChecksNeeded*** is equal to true,

Set *RecordCount* equal to the number of records in ***MatsDhvRecordsByHourLocation*** where ParameterCode is equal to "SO2RE" or "SO2RH".

If ***CurrentHourlyOpRecord***.OperatingTime is greater than 0

If ***MatsSo2ParameterCode*** is null,

If *RecordCount* is greater than 0,

return result A.

Else /* Method Exists for SO2 Surrogate */

If *RecordCount* is equal to 0,

return result E

Else if *RecordCount* is greater than 1,

return result B.

Else /* *RecordCount* is equal to 1 */

Set ***MatsSo2DhvRecord*** to the located ***MatsDhvRecordsByHourLocation*** record.
Append ***MatsSo2DhvRecord***.UnadjustedHourlyValue to ***ApportionmentSo2RateArray***
Append ***MatsSo2DhvRecord***.ModcCode to ***MatsMS1So2ModcCodeArray***

If ***MatsSo2DhvRecord***.ParameterCode is equal to ***MatsSo2ParameterCode***

If ***MatsSo2MethodCode*** is equal to "CALC",

If ***MatsSo2DhvRecord***.EquationCode is null,

If ***MatsHgDhvRecord***.ModcCode is NOT equal to "38",
return result F.
Else
return result H.

Else

If ***MatsSo2DhvRecord***.EquationCode is NOT equal to "MS-1",
return result F.
Else if ***MatsSo2DhvRecord***.ModcCode is NOT equal to "38",
Set ***MatsMs1So2DhvId*** to ***MatsSo2DhvRecord***.MatsDhvId
Set ***MatsMs1So2UnadjustedHrlyValue*** to
MatsSo2DhvRecord.UnadjustedHrlyValue
Set ***MatsParameterPluginSo2*** to ***MatsSo2DhvRecord***.ParameterCode

Else

If ***MatsSo2DhvRecord***.EquationCode is equal to "MS-1",

return result G.

Else

Set ***MatsSo2cNeeded*** to true.

If ***MatsSo2DhvRecord***.ParameterCode is equal to 'SO2RE',

Set ***MatsSo2reDhvChecksNeeded*** to true.

Else if ***MatsSo2DhvRecord***.ParameterCode is equal to 'SO2RH',

Set ***MatsSo2rhDhvChecksNeeded*** to true.

Else /* DHV and Method parameter code mismatch */

return result C.

Else /* Non Operating Hour */

If *RecordCount* is greater than 0,

return result D.

Results:

Result	Response	Severity
A	MATS Derived Hourly Value records were reported for [param], but no supporting method exists.	Critical Error Level 1
B	You reported more than one MATS Derived Hourly Value records for [param] for the hour.	Critical Error Level 1
C	MATS Derived Hourly Value records were reported for [param], but no supporting method exists.	Critical Error Level 1
D	You reported MATS Derived Hourly Value records for [param] that are not reported if the unit did not operate in the hour.	Critical Error Level 1
E	No required MATS Derived Hourly Value records were reported for [param].	Critical Error Level 1
F	The reported MATS method code is equal to "CALC." However, the appropriate Formula Code "MS-1" has not been reported.	Critical Error Level 1
G	The reported Formula Code of "MS-1" is inconsistent with the reported MATS Method Code.	Critical Error Level 1
H	The reported MATS method code is equal to "CALC." However, the appropriate Formula Code "MS-1" has not been reported.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-10

Check Name: MATS Hg: Locate Monitor Hourly Record

Related Former Checks:

Applicability:

Description: Locates the MATS Monitor Hourly record with Parameter Code equal to 'HGC' and for the current hour and location. Returns a negative result if more than one active record is found. Returns the record and a flag indicating whether 'HGC' checks are needed, if one record is found.

Additionally, if the system type of the HGC MHV record is "ST", the check sets FlowMonitorHourlyChecksNeeded to true.

Specifications:

Set ***MatsHgcMhvChecksNeeded*** to false.
Set ***MatsHgcMhvRecord*** to null.

If ***DerivedHourlyChecksNeeded*** is equal to true,

Set *RecordCount* equal to the number of records in ***MatsMhvHgcRecordsByHourLocation***.

If ***CurrentHourlyOpRecord***.OperatingTime is greater than 0,

If ***MatsHgcNeeded*** is equal to false,

If *RecordCount* is greater than 0,

return result A.

Else if *RecordCount* is equal to 0,

return result B.

Else if *RecordCount* is greater than 1,

return result C.

Else /* *RecordCount* is equal to 1 */

Set ***MatsHgcMhvRecord*** to the located ***MatsMhvHgcRecordsByHourLocation*** record.
Set ***MatsHgcMhvChecksNeeded*** to true.

If ***MatsHgcMhvRecord***.SystemTypeCode is equal to "ST",

FlowMonitorHourlyChecksNeeded = true

Else /* Non Operating Hour */

If *RecordCount* is greater than 0,

return result D.

Results:

Result	Response	Severity
A	You reported a MATS Monitor Hourly Value for [param], but did not report either a MATS Derived Hourly Value or monitoring method for that pollutant.	Critical Error Level 1
B	You did not report a MATS Hourly Monitor Value record for [param], though reporting a MATS Derived Hourly Value for the hour.	Critical Error Level 1
C	You reported more than one [param] MATS Monitor Hourly Value for the hour.	Critical Error Level 1
D	You reported a MATS Hourly Monitor Value record for [param] for a non-operating hour.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-11

Check Name: MATS HCl: Locate Monitor Hourly Record

Related Former Checks:

Applicability:

Description: Locates the MATS Monitor Hourly record with Parameter Code equal to 'HCLC' and for the current hour and location. Returns a negative result if more than one active record is found. Returns the record and a flag indicating whether 'HCLC' checks are needed, if one record is found.

Specifications:

Set ***MatsHclcMhvChecksNeeded*** to false.
Set ***MatsHclcMhvRecord*** to null.

If ***DerivedHourlyChecksNeeded*** is equal to true,

Set *RecordCount* equal to the number of records in ***MatsMhvHclcRecordsByHourLocation***.

If ***CurrentHourlyOpRecord***.OperatingTime is greater than 0

If ***MatsHclcNeeded*** is equal to false,

If *RecordCount* is greater than 0,

return result A.

Else if *RecordCount* is equal to 0,

return result B.

Else if *RecordCount* is greater than 1,

return result C.

Else /* *RecordCount* is equal to 1 */

Set ***MatsHclcMhvRecord*** to the located ***MatsMhvHclcRecordsByHourLocation*** record.
Set ***MatsHclcMhvChecksNeeded*** to true.

Else /* Non Operating Hour */

If *RecordCount* is greater than 0,

return result D.

Results:

Result	Response	Severity
A	You reported a MATS Monitor Hourly Value for [param], but did not report either a MATS Derived Hourly Value or monitoring method for that pollutant.	Critical Error Level 1
B	You did not report a MATS Hourly Monitor Value record for [param], though reporting a MATS Derived Hourly Value for the hour.	Critical Error Level 1
C	You reported more than one [param] MATS Monitor Hourly Value for the hour.	Critical Error Level 1
D	You reported a MATS Hourly Monitor Value record for [param] for a non-operating hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-12

Check Name: MATS HF: Locate Monitor Hourly Record

Related Former Checks:

Applicability:

Description: Locates the MATS Monitor Hourly record with Parameter Code equal to 'HFC' and for the current hour and location. Returns a negative result if more than one active record is found. Returns the record and a flag indicating whether 'HFC' checks are needed, if one record is found.

Specifications:

Set ***MatsHfcMhvChecksNeeded*** to false.
Set ***MatsHfcMhvRecord*** to null.

If ***DerivedHourlyChecksNeeded*** is equal to true,

Set *RecordCount* equal to the number of records in ***MatsMhvHfcRecordsByHourLocation*** where ParameterCode is equal to "HFC".

If ***CurrentHourlyOpRecord***.OperatingTime is greater than 0

If ***MatsHfcNeeded*** is equal to false,

If *RecordCount* is greater than 0,

return result A.

Else if *RecordCount* is equal to 0,

return result B.

Else if *RecordCount* is greater than 1,

return result C.

Else /* *RecordCount* is equal to 1 */

Set ***MatsHfcMhvRecord*** to the located ***MatsMhvHfcRecordsByHourLocation*** record.
Set ***MatsHfcMhvChecksNeeded*** to true.

Else /* Non Operating Hour */

If *RecordCount* is greater than 0,

return result D.

Results:

Result	Response	Severity
A	You reported a MATS Monitor Hourly Value for [param], but did not report either a MATS Derived Hourly Value or monitoring method for that pollutant.	Critical Error Level 1
B	You did not report a MATS Hourly Monitor Value record for [param], though reporting a MATS Derived Hourly Value for the hour.	Critical Error Level 1
C	You reported more than one [param] MATS Monitor Hourly Value for the hour.	Critical Error Level 1
D	You reported a MATS Hourly Monitor Value record for [param] for a non-operating hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-13

Check Name: MATS: Check MATS Load

Related Former Checks:

Applicability:

Description: Enusres that the MATS Load is reported when the current hour is operating and an active HGRE, HCLRE, HFRE or SO2RE method exists.

Specifications:

Set ***ApportionmentMatsLoadArray*** for this Location to null

If ***CurrentHourlyOpRecord*** is not null

If ***CurrentHourlyOpRecord***.OperatingTime is greater than 0,

/* Count the number of RE methods active during the hour and associates with a location in the monitoring plan */
Count the record in ***MonitorMethodRecordsByHour*** where ParameterCode is equal to "HGRE", "HCLRE", "HFRE", or "SO2RE",

If the count is greater than 0,

Set ***ApportionmentMatsLoadArray*** for this Location to ***CurrentHourlyOpRecord***.MatsHourLoad.

If ***CurrentHourlyOpRecord***.MatsHourLoad is null,

return result A

Else if ***CurrentHourlyOpRecord***.LoadUnitsOfMeasureCode = "MW" AND ***CurrentHourlyOpRecord***.MatsHourLoad is less than ***CurrentHourlyOpRecord***.HourLoad

If ***MpStackConfigForHourlyChecks*** is NOT equal to "MS",

return result D

Else

If ***CurrentHourlyOpRecord***.MatsHourLoad is not null,

return result B.

Else

If ***CurrentHourlyOpRecord***.MatsHourLoad is not null,

return result C.

Results:

Result	Response	Severity
A	You did not provide a MATSHourLoad record, which is required when you report an output based emission rate.	Critical Error Level 1
B	You provided a MATSHourLoad record which is not required when you report a heat input based emission rate.	Informational Message
C	You reported a MATSHourLoad record, but this is not appropriate for a non-operating hour.	Critical Error Level 1
D	The reported MATSHourLoad is less than the reported HourLoad value.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-14

Check Name: Update Sorbent Trap Operating Date List

Related Former Checks:

Applicability:

Description: For operating hours, this check inserts the current date into OperatingDateList for the current location's sorbent traps that are active for the current hour.

Specifications:

If ***CurrentHourlyOpRecord*** is NOT null,

For each entry in ***MatsSorbentTrapListByLocationArray*** where the array index is ***CurrentMonitorPlanLocationPosition***,

When:

1) ***CurrentHourlyOpRecord***.OperatingTime is greater than 0.
2) ***CurrentOperatingDate*** is on or after the date of the entry's SorbentTrapBeginDatehour.
3) ***CurrentOperatingDate*** is on or before the date of the entry's SorbentTrapEndDatehour.
4) ***CurrentOperatingDate*** is not in the entry's OperatingDateList.

Then:

Append ***CurrentOperatingDate*** to the entry's OperatingDateList.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Code: MATSHOD-15

Check Name: Initialize Message Plug-ins

Related Former Checks:

Applicability:

Description: Initializes the plugin parameters.

Specifications:

Set ***MatsHclDhvParameterDescription*** to "HCLRE or HCLRH".
Set ***MatsHclMhvParameterDescription*** to "HCLC".
Set ***MatsHfDhvParameterDescription*** to "HFRE or HFRH".
Set ***MatsHfMhvParameterDescription*** to "HFC".
Set ***MatsHgDhvParameterDescription*** to "HGRE or HGRH".
Set ***MatsHgMhvParameterDescription*** to "HGC".
Set ***MatsSo2DhvParameterDescription*** to "SO2RE or SO2RH".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report Summary Value Initialization

Check Code: MATSHOD-16

Check Name: Verify Hourly GFM for Active Sampling Trains

Related Former Checks:

Applicability:

Description: Verifies that a GFM exists for each passed, failed or uncertain sampling trains active for a particular operating hour. Allows GFM records for any operating hour as long as an associated sampling trains exists, but does not allow GFM for non operating hours.

Specifications:

Set ***MatsMissingGfmList*** to "".
Set ***MatsMultipleGfmList*** to "".

If ***DerivedHourlyChecksNeeded*** is equal to true,

If ***CurrentHourlyOpRecord***.OperatingTime is greater than 0

If ***MatsHgcMhvRecord*** is null, OR ***MatsHgcMhvRecord***.ModcCode is NOT equal to "41" or "42",

Set *LocatedMatsSamplingTrainRecords* to the records in ***MatsSamplingTrainRecords*** where:

1) LocationId is equal to ***CurrentHourlyOpRecord***.LocationId
2) SorbentTrapBeginDateHour is on or before ***CurrentDateHour***
3) SorbentTrapEndDateHour is on or after ***CurrentDateHour***

For each *MatsSamplingTrainRecord* in *LocatedMatsSamplingTrainRecords*,

Count the number of ***MatsHourlyGfmRecordsForHourAndLocation*** where ComponentId is equal to *MatsSamplingTrainRecord*.ComponentId.

If the count is equal to 0,

If *MatsSamplingTrainRecord*.TrainQAStatus is equal to "PASSED" or "UNCERTAIN", AND ***MatsHgcMhvRecord*** is null OR ***MatsHgcMhvRecord***.ModcCode is NOT equal to "34",

If ***MatsHgcMhvRecord*** is NOT null, AND ***MatsHgcMhvRecord***.ModcCode is equal to "32",

Count the number of *LocatedMatsSamplingTrainRecords* where:

1) ComponentId is equal to *MatsSamplingTrainRecord*.ComponentId.
2) TrapModcCode is equal to "32".
3) TrainQAStatus is NOT equal to "PASSED" or "UNCERTAIN".

If the count is equal to 0,
Append *MatsSamplingTrainRecord*.Description to ***MatsMissingGfmList***.

Else
Append *MatsSamplingTrainRecord*.Description to ***MatsMissingGfmList***.

Else if the count is greater than 1,

Append *MatsSamplingTrainRecord*.Description to ***MatsMultipleGfmList***.

If both ***MatsMissingGfmList*** and ***MatsMultipleGfmList*** are NOT empty,

return result A.

Else if ***MatsMissingGfmList*** is NOT empty,

return result B.

Else if ***MatsMultipleGfmList*** is NOT empty,

return result C.

Else

Count the number of ***MatsHourlyGfmRecordsForHourAndLocation***.

If the count is greater than 0,

return result D.

Results:

Result	Response	Severity
A	For the current hour, GFM records are missing for "PASSED" or "UNCERTAIN" sampling train(s) [Missing], and multiple GFM records exist for sampling train(s) [Multiple].	Informational Message
B	For the current hour, GFM records are missing for "PASSED" or "UNCERTAIN" sampling train(s) [Missing].	Informational Message
C	For the current hour, multiple GFM records exist for sampling train(s) [Multiple].	Informational Message
D	You reported a GFM record for a non-operating hour, which is not appropriate.	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ Operating Hour Evaluation

Check Category:

MATS Sampling Train Checks

Check Code: MATSTRN-1

Check Name: Component ID Valid

Related Former Checks:

Applicability:

Description: Ensure that sampling train component id exists and that the associated component type is "STRAIN".

Specifications:

Set ***MatsSamplingTrainComponentIdValid*** equal to false.

If ***MatsSamplingTrainRecord***.ComponentID is null,

Set ***MatsSamplingTrainProblemComponentExists*** equal to true.
Return result A.

Else if ***MatsSamplingTrainRecord***.ComponentTypeCode is not equal to "STRAIN",

Set ***MatsSamplingTrainProblemComponentExists*** equal to true.
Return result B.

Else

Set ***MatsSamplingTrainComponentIdValid*** equal to true.

Add an entry to ***MatsSamplingTrainDictionary*** with a key equal to ***MatsSamplingTrainRecord***.TrainID and the value record initialized with the following values:

1) Set HgConcentration equal to null
2) Set TrainQAStatusCode equal to null
3) Set ReferenceSFSRRatio equal to null
4) Set TotalSFSRRatioCount equal to ***MatsSamplingTrainRecord***.SfsrTotalCount with a default of 0 when ***MatsSamplingTrainRecord***.SupplementalDataInd is equal to 1. Otherwise set to 0.
5) Set DeviatedSFSRRationCount equal to ***MatsSamplingTrainRecord***.SfsrDeviatedCount with a default of 0 when ***MatsSamplingTrainRecord***.SupplementalDataInd is equal to 1. Otherwise set to 0.
6) Set TotalGfmCount equal to ***MatsSamplingTrainRecord***.GfmTotalCount with a default of 0 when ***MatsSamplingTrainRecord***.SupplementalDataInd is equal to 1. Otherwise set to 0.
7) Set NotAvailableGfmCount equal to ***MatsSamplingTrainRecord***.GfmNotAvailableCount with a default of 0 when ***MatsSamplingTrainRecord***.SupplementalDataInd is equal to 1. Otherwise set to 0.
8) Set SamplingTrainValid equal to true
9) Set IsBorderTrap to (***MatsSamplingTrainRecord***.BorderTrapIndicator is equal to 1).
10) Set IsSupplementalData to (***MatsSamplingTrainRecord***.SupplementalDataIndicator is equal to 1).

Add the same entry to ***MatsSorbentTrapSamplingTrainList***.

Results:

Result	Response	Severity
A	For [key], you have not reported a value for [fieldname], which is required.	Critical Error Level 1
B	The [fieldname] in the monitoring plan is [component type]. A [component type] [fieldname] is not associated with sorbent trap data.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Initialization

Check Code: MATSTRN-2

Check Name: Sorbent Trap Serial Number

Related Former Checks:

Applicability:

Description: Check that a sorbent trap serial number is provided.

Specifications:

If the ***MatsSamplingTrainRecord***.SorbentTrapSn is null,

Return result A.

Results:

Result	Response	Severity
A	You did not provide a [fieldname], which is required, for [key].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-3

Check Name: Train Quality Assurance Status Valid

Related Former Checks:

Applicability:

Description: Check Sampling Train Quality Assurance Status Matches Lookup Table

Validation Tables:

Train Qa Status Code (Lookup Table)

Specifications:

Set ***MatsSamplingTrainQaStatusCodeValid*** to false.

If ***MatsSamplingTrainRecord***.TrainQAStatusCode is null,

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result A,

Else if ***MatsSamplingTrainRecord***.TrainQAStatusCode does not match a value in ***MatsSamplingTrainQaStatusLookupTable***,

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result B.

Else

Set ***MatsSamplingTrainQaStatusCodeValid*** to true.

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.TrainQAStatusCode to ***MatsSamplingTrainRecord***.TrainQAStatusCode where the key equals ***MatsSamplingTrainRecord***.TrainID.

Results:

Result	Response	Severity
A	For [key], you have not reported a value for [fieldname], which is required.	Critical Error Level 1
B	For [key] you reported a [value] which is not valid for [fieldname].	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-4

Check Name: Main Trap Hg Valid

Related Former Checks:

Applicability:

Description: Main Trap Hg Null or Reported to Two Decimal Places

Specifications:

Set ***MatsMainTrapHgValid*** to false.

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If the ***MatsSamplingTrainRecord***.MainTrapHg is null,

If the ***MatsSamplingTrainRecord***.QAStatusCode is not "INC", "EXPIRED", or "LOST",

Return result A.

Otherwise

Set ***MatsMainTrapHgValid*** to true.

Else,

If the ***MatsSamplingTrainRecord***.QAStatus Code is not "PASSED", "FAILED", or "UNCERTAIN",

Return result B.

Else, if the ***MatsSamplingTrainRecord***.MainTrapHg is not reported in scientific notation rounded to three decimal places, keeping one to the left of the decimal point (x.xx-E-x),

Return result C.

Otherwise

Set ***MatsMainTrapHgValid*** to true.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	The [fieldname] value in the [key] records is not reported in scientific notation rounded to three significant figures, with one digit to the left of the decimal point.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-5

Check Name: BT Trap Hg Valid

Related Former Checks:

Applicability:

Description: BT Trap Hg Null or Reported to Two Decimal Places

Specifications:

Set ***MatsBtTrapHgValid*** to false.

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If the ***MatsSamplingTrainRecord.***BTTrapHg is null,

If the ***MatsSamplingTrainRecord***.QAStatusCode is not "INC", "EXPIRED", or "LOST",

Return result A.

Otherwise

Set ***MatsBtTrapHgValid*** to true.

Else,

If the ***MatsSamplingTrainRecord***.QAStatus Code is not "PASSED", "FAILED", or "UNCERTAIN",

Return result B.

Else if the ***MatsSamplingTrainRecord***.BTTrapHg is not reported in scientific notation rounded to three decimal places, keeping one to the left of the decimal point (x.xx-E-x),

Return result C.

Otherwise

Set ***MatsBtTrapHgValid*** to true.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	The [fieldname] value in the [key] records is not reported in scientific notation rounded to three significant figures, with one digit to the left of the decimal point.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-6

Check Name: Spike Trap Hg Valid

Related Former Checks:

Applicability:

Description: Spike Trap Hg Null or Reported to Two Decimal Places

Specifications:

Set ***MatsSpikeTrapHgValid*** to false.

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If the ***MatsSamplingTrainRecord***.SpikeTrapHg is null,

If the ***MatsSamplingTrainRecord***.QAStatusCode is not "INC", "EXPIRED", or "LOST",

Return result A.

Otherwise

Set ***MatsSpikeTrapHgValid*** to true.

Else,

If the ***MatsSamplingTrainRecord***.QAStatus Code is not "PASSED", "FAILED", or "UNCERTAIN",

Return result B.

Else if the ***MatsSamplingTrainRecord***.SpikeTrapHg is not reported in scientific notation rounded to three decimal places, keeping one to the left of the decimal point (x.xx-E-x),

Return result C.

Otherwise

Set ***MatsSpikeTrapHgValid*** to true.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	The [fieldname] value in the [key] records is not reported in scientific notation rounded to three significant figures, with one digit to the left of the decimal point.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-7

Check Name: Spike Reference Value Valid

Related Former Checks:

Applicability:

Description: Spike Reference Value Null or Reported to Two Decimal Places

Specifications:

Set ***MatsSpikeReferenceValueValid*** to false.

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If the ***MatsSamplingTrainRecord***.SpikeReferenceValue is null,

If the ***SamplingTrainData***.QAStatusCode is not "INC", "EXPIRED", or "LOST",

Return result A.

Otherwise

Set Mats***SpikeReferenceValueValid*** to true.

Else,

If the ***MatsSamplingTrainRecord***.QAStatus Code is not "PASSED", "FAILED", or "UNCERTAIN",

Return result B.

Else if the ***MatsSamplingTrainRecord***.SpikeReferenceValue is not reported in scientific notation rounded to three decimal places, keeping one to the left of the decimal point (x.xx-E-x),

Return result C.

Otherwise

Set Mats***SpikeReferenceValueValid*** to true.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	The [fieldname] value in the [key] records is not reported in scientific notation rounded to three significant figures, with one digit to the left of the decimal point.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-8

Check Name: Total Sample Volume DSCM Valid

Related Former Checks:

Applicability:

Description: Total Sample Volume DSCM Null or Reported to Two Decimal Places

Specifications:

Set ***MatsTotalSampleVolumeDSCMValid*** to false.

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If the ***MatsSamplingTrainRecord***.TotalSampleVolumeDSCM is null,

If the ***SamplingTrainData***.QAStatusCode is not "INC", "EXPIRED", or "LOST",

Return result A.

Otherwise

Set ***MatsTotalSampleVolumeDSCMValid*** to true.

Else,

If the ***MatsSamplingTrainRecord***.QAStatus Code is not "PASSED", "FAILED", or "UNCERTAIN",

Return result B.

Else if the ***MatsSamplingTrainRecord***.TotalSampleVolumeDSCM is less than than two decimal places ,

Return result C.

Otherwise

Set ***MatsTotalSampleVolumeDSCMValid*** to true.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	For [key], the [fieldname] value must be reported to at least two decimal places.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-9

Check Name: Reference SFSR Ratio Valid

Related Former Checks:

Applicability:

Description: Reference SFSR Ratio Null or Reported to Two Decimal Places

Specifications:

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If the ***MatsSamplingTrainRecord***.ReferenceSFSRRatio is null,

If the ***MatsSamplingTrainRecord***.QAStatusCode is not "INC", "EXPIRED", or "LOST", AND the ***MatsSamplingTrainRecord***.RATAIndicator is NOT equal to 1,

If ***MatsSamplingTrainDictionary*** contains a lookup key equal to ***MatsSamplingTrainRecord***.TrainID,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result A.

Else,

If the ***MatsSamplingTrainRecord***.QAStatusCode is not "PASSED", "FAILED", or "UNCERTAIN",

If ***MatsSamplingTrainDictionary*** contains a lookup key equal to ***MatsSamplingTrainRecord***.TrainID,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result B.

Else if the ***MatsSamplingTrainRecord***.ReferenceSFSRRatio is not reported to one decimal place,

If ***MatsSamplingTrainDictionary*** contains a lookup key equal to ***MatsSamplingTrainRecord***.TrainID,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result C.

Else if the ***MatsHourlyGFMRecord.***HourlySFSRRatio is not greater than or equal to 1.0 and less than or equal to 100.0,

If ***MatsSamplingTrainDictionary*** contains a lookup key equal to ***MatsSamplingTrainRecord***.TrainID,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result D.

Else

If ***MatsSamplingTrainDictionary*** contains a lookup key equal to ***MatsSamplingTrainRecord***.TrainID,
Set ***MatsSamplingTrainDictionary***.ReferenceSFSRRatio to ***MatsSamplingTrainRecord***.ReferenceSFSRRatio where the key equals ***MatsSamplingTrainRecord***.TrainID.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	The [fieldname] value for [key] should be reported to one decimal place.	Critical Error Level 1
D	The [fieldname] value for [key] must be a number between 1 and 100.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-10

Check Name: Sampling Ratio Check Result Code Valid

Related Former Checks:

Applicability:

Description: Sampling Ratio Check Result Code Valid

Specifications:

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If ***MatsSamplingTrainRecord***.SamplingRatioCheckResultCode is null,

If the ***MatsSamplingTrainRecord***.QAStatusCode is not "INC", "EXPIRED", or "LOST",

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result A.

Else,

If ***MatsSamplingTrainRecord***.SamplingRatioCheckResultCode is equal to "PASSED",

If ***MatsSamplingTrainRecord***.QAStatus Code is not equal "PASSED", "FAILED", or "UNCERTAIN",

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result B.

Else if ***MatsSamplingTrainRecord***.SamplingRatioCheckResultCode is equal to "FAILED",

If ***MatsSamplingTrainRecord***.QAStatus Code is not equal "FAILED",

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result C.

Otherwise

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result D.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	For [key], you reported that the sampling train SFSR Ratio check FAILED, but did not also report the train QA Status Code as FAILED.	Critical Error Level 1
D	For [key], the [fieldname] is not reported as PASSED or FAILED..	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-11

Check Name: Post Leak Check Result Code Valid

Related Former Checks:

Applicability:

Description: Post Leak Check Result Code Valid

Specifications:

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If the ***MatsSamplingTrainRecord***.PostLeakCheckResultCode is null,

If the ***MatsSamplingTrainRecord***.QAStatusCode is not "INC", "EXPIRED" or "LOST",

Return result A.

Else

If ***MatsSamplingTrainRecord***.PostLeakCheckResultCode is equal to "PASSED",

If ***MatsSamplingTrainRecord***.QAStatus Code is not equal to "PASSED", "FAILED", OR "UNCERTAIN",

Return result B.

Else if ***MatsSamplingTrainRecord***.PostLeakCheckResultCode is equal to "FAILED",

If ***MatsSamplingTrainRecord***.QAStatus Code is not equal to "FAILED",

Return result C.

Otherwise

Return result D.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	The [fieldname] value of [value] from the [key] records exceeds the PS-12B breakthrough criteria, but you did not report the train QA Status Code as FAILED.	Critical Error Level 1
D	The [fieldname] is not reported as PASSED or FAILED.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-12

Check Name: Sample Damage Explanation

Related Former Checks:

Applicability:

Description: Sample Damage Explanation is provided if QA Status Code equals LOST.

Specifications:

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If the ***MatsSamplingTrainRecord***.SampleDamageExplanation is null,

If the ***MatsSamplingTrainRecord***.QAStatus Code is equal to "LOST",

Return result A.

Results:

Result	Response	Severity
A	For [key], you did not report a SampleDamageExplanation which is required if the sorbent train QA Status Code is LOST.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-13

Check Name: Hg Concentration reported properly

Related Former Checks:

Applicability:

Description: Hg Concentration is reported properly

Specifications:

Set ***MatsCalcTrainHgConcentration*** = null.

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If ***MatsSamplingTrainRecord***.HgConcentration is null,

If ***MatsSamplingTrainRecord***.TrainQAStatusCode is not "INC", "EXPIRED", or "LOST",

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result A.

Else,

If ***MatsSamplingTrainRecord***.TrainQAStatusCode is not "PASSED", "FAILED", or "UNCERTAIN",

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result B.

Else if ***MatsSamplingTrainRecord***.HgConcentration is not reported in scientific notation rounded to three decimal places, keeping one to the left of the decimal point (x.xx-E-x).

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result C.

Else if ***MatsSamplingTrainRecord.***ModcCode is 43 or 44,

Set ***MatsCalcTrainHgConcentration*** = ***MatsSamplingTrainRecord.***HgConcentration**.**

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.HgConcentration to ***MatsSamplingTrainRecord.***HgConcentration where the key equals ***MatsSamplingTrainRecord***.TrainID.

Else if ***MatsMainTrapHgValid*** is true AND ***MatsBTTrapHgValid*** is true AND ***MatsTotalSampleVolumeDSCMValid*** is true,

Set ***MatsCalcTrainHgConcentration*** = (***MatsSamplingTrainRecord***.MainTrapHg + ***MatsSamplingTrainRecord***.BTTrapHg) / ***MatsSamplingTrainRecord***.TotalSampleVolumeDSCM, rounded to three significant figures, using scientific notation (x.xx-E-2).

If ***MatsSamplingTrainComponentIdValid*** is true,

Set ***MatsSamplingTrainDictionary***.HgConcentration to ***MatsCalcTrainHgConcentration*** where the key equals ***MatsSamplingTrainRecord***.TrainID.

If ***MatsSamplingTrainRecord***.HgConcentration is not equal to ***MatsCalcTrainHgConcentration***,

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Return result D

Else // A calculation input is not valid

If ***MatsSamplingTrainComponentIdValid*** is true,
Set ***MatsSamplingTrainDictionary***.SamplingTrainValid to false where the key equals ***MatsSamplingTrainRecord***.TrainID.

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	The [fieldname] value in the [key] records is not reported in scientific notation rounded to three significant figures, with one digit to the left of the decimal point.	Critical Error Level 1
D	The [fieldname] is inconsistent with the value [value] calculated from the reported [key] records used in the calculation.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-14

Check Name: Percent Breakthrough reported properly.

Related Former Checks:

Applicability:

Description: Percent Breakthrough is reported properly. Includes alternate criteria for RATAs

Specifications:

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If the ***MatsSamplingTrainRecord***.PercentBreakthrough is null,

If the ***MatsSamplingTrainRecord***.QAStatus Code is "PASSED", "FAILED", or "UNCERTAIN", AND the. ***MatsSamplingTrainRecord***.HgConcentration is greater than or equal to 0.2,

Return result A.

Else,

If the ***MatsSamplingTrainRecord***.QAStatusCode is “LOST”, “EXPIRED”, or “INC”,

Return result B

Else if the ***MatsSamplingTrainRecord***.PercentBreakthrough is NOT reported to one decimal place.

Return result C

Else if ***MatsMainTrapHgValid*** is equal to true, and ***MatsBtTrapHgValid*** is equal to true,

Set ***MatsCalcTrainPercentBreakthrough*** = (***MatsSamplingTrainRecord***.BTTrapHg / ***MatsSamplingTrainRecord***.MainTrapHg) x 100, rounded to one decimal place.

If ***MatsSamplingTrainRecord***.PercentBreakthrough is NOT equal to ***MatsCalcTrainPercentBreakthrough***,

Return result D,

Else,

If the ***MatsSamplingTrainRecord***.SorbentTrapApsCode is equal to 'RATA',

If ***MatsSamplingTrainRecord***.HgConcentration is greater than 1 AND
MatsSamplingTrainRecord.PercentBreakthrough rounded to an integer is greater than 10%, OR
MatsSamplingTrainRecord.HgConcentration is greater than 0.5 AND
MatsSamplingTrainRecord.PercentBreakthrough rounded to an integer is greater than 20%, OR
MatsSamplingTrainRecord.HgConcentration is greater than 0.1 AND
MatsSamplingTrainRecord.PercentBreakthrough rounded to an integer is greater than 50%,

If ***MatsSamplingTrainRdcord***.TrainQAStatusCode is NOT equal to "FAILED',

Return result F.

Else

If the ***MatsSamplingTrainRecord***.HgConcentration is NOT less than 0.2,

If The***MatsSamplingTrainRecord***.PercentBreakthrough rounded to an integer is greater than 10%, OR
the ***MatsSamplingTrainRecord***.PercentBreakthrough rounded to an integer is greater

than 5%, AND the ***MatsSamplingTrainRecord***.HgConcentration is greater than 0.5,

If ***MatsSamplingTrainRecord***.TrainQAStatusCode is NOT equal to "FAILED",

Return result E

Results:

Result	Response	Severity
A	For [key], you did not report a [fieldname] value which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the train Hg Concentration is not less than 10% of the Hg limit equivalent concentration or less than or equal to 0.1 ug/dscm if performing a RATA.	Critical Error Level 1
B	For [key], you reported a [fieldname] value which is not reported if the sorbent train QA Status Code is LOST, EXPIRED, or INC.	Critical Error Level 1
C	The [fieldname] value for [key] should be reported to one decimal place.	Critical Error Level 1
D	The [fieldname] is inconsistent with the value [value] calculated from the reported [key] records used in the calculation.	Critical Error Level 1
E	The [fieldname] value of [value] from the [key] records exceeds the PS-12B breakthrough criteria, but you did not report the train QA Status Code as FAILED.	Critical Error Level 1
F	You reported a Sampling Train Data QA Status of PASSED or UNCERTAIN, but at least one of the alternative performance specifications listed in Section 4.1.2.2 in Appendix A CFR Part 63 to PS12B was not met.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-15

Check Name: Percent Spike Recovery reported properly

Related Former Checks:

Applicability:

Description: Percent Spike Recovery reported properly

Specifications:

If ***MatsSamplingTrainQaStatusCodeValid*** is true,

If the ***MatsSamplingTrainRecord***.PercentSpikeRecovery is null,

If the ***MatsSamplingTrainRecord***.QAStatus Code is not "INC", "EXPIRED", or "LOST",

Return result A.

Else

If the ***MatsSamplingTrainRecord***.QAStatus Code is not "PASSED", "FAILED", or "UNCERTAIN",

Return result B.

Else if the ***MatsSamplingTrainRecord***.PercentSpikeRecovery is not reported to one decimal place,

Return result C.

Else if ***MatsSpikeTrapHgValid*** is equal to true, AND ***Mats*****S*****pikeReferenceValueValid,*** is equal to true,

Set ***MatsCalcTrainPercentSpikeRecovery*** = ***MatsSamplingTrainRecord***.SpikeTrapHg / ***MatsSamplingTrainRecord***.SpikeReferenceValue) x 100, rounded to one decimal place.

If ***MatsSamplingTrainRecord***.PercentSpikeRecovery is not equal to ***MatsCalcTrainPercentSpikeRecovery***,

Return result D.

Else if the ***MatsSamplingTrainRecord***.PercentSpikeRecovery is less than 75% or greater than 125%

If ***MatsSamplingTrainRecord***.TrainQAStatusCode is not equal to "FAILED",

Return result E

Results:

Result	Response	Severity
A	You did not report a [fieldname] value in the [key] records which is required if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
B	You reported a [fieldname] value in the [key] records which is reported only if the sorbent train QA Status Code is PASSED, FAILED, or UNCERTAIN, and the stack flow rate for the hour is a measured data value.	Critical Error Level 1
C	The [fieldname] value for [key] should be reported to one decimal place.	Critical Error Level 1
D	The [fieldname] is inconsistent with the value [value] calculated from the reported [key] records used in the calculation.	Critical Error Level 1
E	The [fieldname] value of [value] from the [key] records exceeds the PS-12B breakthrough criteria, but you did not report the train QA Status Code as FAILED.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sampling Train Evaluation

Check Code: MATSTRN-16

Check Name: Check Hourly Sampling Ratios

Related Former Checks:

Applicability:

Description: Compare Hourly Sampling Ratio with PS12B Requirement

Specifications:

If ***MatsSamplingTrainDictionary*** contains the key ***MatsSamplingTrainRecord***.TrainID,
Set *SamplingTrainValid* to ***MatsSamplingTrainDictionary***.SamplingTrainValid where the key equals ***MatsSamplingTrainRecord***.TrainID.
Else
Set *SamplingTrainValid* to false.

If *SamplingTrainValid* is true,

Set *TotalSFSRRatioCount* to ***MatsSamplingTrainDictionary***.TotalSFSRRatioCount where the key equals ***MatsSamplingTrainRecord***.TrainID.
Set *DeviatedSFSRRatioCount* to ***MatsSamplingTrainDictionary***.DeviatedSFSRRatioCount where the key equals ***MatsSamplingTrainRecord***.TrainID.
Set *SamplingTrainCountsAreComplete* to (***MatsSamplingTrainDictionary***.IsBorderTrain is false OR ***MatsSamplingTrainDictionary***.IsSupplementalData is true).

If *TotalSFSRRatioCount* is greater than or equal to 100,

Set *MatsCalcPercentSFSRRatioDev* to *DeviatedSFSRRatioCount* / *TotalSFSRRatioCount* x 100**,** rounded to an integer.

If the ***MatsSamplingTrainRecord***.SamplingRatioCheckResultCode is equal to "PASSED"

If the *MatsCalcPercentSFSRRatioDev* is greater than 5,

Return result A.

Else // *TotalSFSRRatioCount* is less than 100

If the ***MatsSamplingTrainRecord***.SamplingRatioCheckResultCode is equal to "PASSED"

If the *DeviatedSFSRRatioCount* is greater than 5,

Return result C.

Results:

Result	Response	Severity
A	For [key], you reported that the SFSR Ratio Check PASSED, but more than five percent of hourly SFSR Ratios deviated from the reference ratio by more than 25 percent.	Critical Error Level 1
B	For [key], you reported that SFSR Ratio Check FAILED, but not more than five percent of hourly SFSR Ratios deviated from the reference ratio by more than 25 percent.	Critical Error Level 1
C	For [key], you reported that the SFSR Ratio Check PASSED, but more than five hourly SFSR Ratios deviated from the reference ratio by more than 25 percent.	Critical Error Level 1
D	For [key], you reported that the SFSR Ratio Check FAILED, but not more than five hourly SFSR Ratios deviated from the reference ratio by more than 25 percent.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ MATS Sampling Train Last Hour Evaluation

Check Code: MATSTRN-17

Check Name: Check the Not Available GFM count compared to the Total GFM Count

Related Former Checks:

Applicability:

Description: Checks the percentage of GFM with an "N" (Not Allowed) Begin End Flag.

Specifications:

If ***MatsSamplingTrainRecord***.TrainQAStatusCode is equal to "PASSED", "FAILED" or "UNCERTAIN",
AND ***MatsSamplingTrainRecord***.RataInd is equal to 0 (zero) or null,

If ***MatsSamplingTrainDictionary*** contains the key ***MatsSamplingTrainRecord***.TrainID,

Set *DictionaryEntry* to ***MatsSamplingTrainDictionary*** where the key equals ***MatsSamplingTrainRecord***.TrainID.

If *DictionaryEntry*.SamplingTrainValid is true,

If *DictionaryEntry*.TotalGfmCount is greater than 0, AND *DictionaryEntry*.NotAvailableGfmCount is greater than or equal to 0,

Set *NotAvailableGfmPercent* to 100 * *DictionaryEntry*.NotAvailableGfmCount / *DictionaryEntry*.TotalGfmCount.

If *NotAvailableGfmPercent* is greater than or equal to 20%,

Return result A.

Results:

Result	Response	Severity
A	For sampling train [key], at least 20 percent of the gas flow meter hours reported a Begin-End Flag of "N".	Informational Message

Usage:

1 Process/Category: Emissions Data Evaluation Report ------ MATS Sampling Train Last Hour Evaluation

Check Category:

MATS Sorbent Trap Data

Check Code: MATSTRP-1

Check Name: Begin Date Valid

Related Former Checks:

Applicability:

Description: This check determines if the sorbent trap data begin date is valid.

Specifications:

Set ***MatsSorbentTrapBeginDateValid*** equal to false.

If the ***MatsSorbentTrapRecord***.BeginDate is null,

Set ***MatsSorbentTrapEvaluationNeeded*** to false.
Return Result A.

Else

Set ***MatsSorbentTrapBeginDateValid*** equal to true.

Results:

Result	Response	Severity
A	For [key], a value for [fieldname] is required.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- MATS Sorbent Trap Hour and Range Evaluation

Check Code: MATSTRP-2

Check Name: Begin Hour Valid

Related Former Checks:

Applicability:

Description: This check determines if the sorbent trap data begin hour is valid

Specifications:

Set ***MatsSorbentTrapBeginDateHourValid*** equal to false.

If ***MatsSorbentTrapBeginDateValid***,

If the ***MatsSorbentTrapRecord***.BeginHour is null,

Set ***MatsSorbentTrapEvaluationNeeded*** to false.
Return Result A.

Else, if the ***MatsSorbentTrapRecord***.BeginHour is less than 0 or greater than 23,

Set ***MatsSorbentTrapEvaluationNeeded*** to false.
Return Result B.

Else

MatsSorbentTrapBeginDateHourValid equal to true.

Results:

Result	Response	Severity
A	For [key], a value for [fieldname] is required.	Critical Error Level 1
B	For [key], you have reported a Begin Hour not between 0 and 23.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --- MATS Sorbent Trap Hour and Range Evaluation

Check Code: MATSTRP-3

Check Name: End Date Valid

Related Former Checks:

Applicability:

Description: This check determines if the sorbent trap data end date and hour is valid

Specifications:

Set ***MatsSorbentTrapEndDateValid*** equal to false.

If the ***MatsSorbentTrapRecord***.EndDate is null,

Set ***MatsSorbentTrapEvaluationNeeded*** to false.
Return Result A.

Else

Set ***MatsSorbentTrapEndDateValid*** equal to true.

Results:

Result	Response	Severity
A	For [key], a value for [fieldname] is required.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- MATS Sorbent Trap Hour and Range Evaluation

Check Code: MATSTRP-4

Check Name: End Hour Valid

Related Former Checks:

Applicability:

Description: This check determines if the sorbent trap data end hour is valid

Specifications:

Set ***MatsSorbentTrapEndDateHourValid*** equal to false.

If the ***MatsSorbentTrapRecord***.EndHour is null,

Set ***MatsSorbentTrapEvaluationNeeded*** to false.
Return Result A.

Else, if the ***MatsSorbentTrapRecord***.EndHour is less than 0 or greater than 23,

Set ***MatsSorbentTrapEvaluationNeeded*** to false.
Return Result B.

Else

Set ***MatsSorbentTrapEndDateHourValid*** equal to true.

Results:

Result	Response	Severity
A	For [key], a value for [fieldname] is required.	Critical Error Level 1
B	For [key], you have reported an End Hour that is not between 0 and 23.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --- MATS Sorbent Trap Hour and Range Evaluation

Check Code: MATSTRP-5

Check Name: Begin and End Times Consistent

Related Former Checks:

Applicability:

Description: Check that the Sorbent Trap end date and time occurs after the begin date and time.

Specifications:

Set ***MatsSorbentTrapDatesAndHoursConsistent*** to false.

If ***MatsSorbentTrapBeginDateHourValid*** is true AND ***MatsSorbentTrapEndDateHourValid*** is true,

If the the ***MatsSorbentTrapRecord***,BeginDateHour is after the ***MatsSorbentTrapRecord***.EndDateHour,

Set ***MatsSorbentTrapEvaluationNeeded*** to false.
Return result A.

Else

Set ***MatsSorbentTrapDatesAndHoursConsistent*** to true.

Results:

Result	Response	Severity
A	For [key], the BeginDate/Hour is inconsistent with the EndDate/Hour.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --- MATS Sorbent Trap Hour and Range Evaluation

Check Code: MATSTRP-6

Check Name: Check For Overlap With Another Sorbent Trap

Related Former Checks:

Applicability:

Description: Check for overlap with the last Sorbent Trap from the previous emission report or with another Sorbent Trap reported in the current emission report.

Specifications:

Locate ***MatsSorbentTrapRecords*** where:

1) SystemId equals ***MatsSorbentTrapRecord***.SystemId
2) TrapId does not equal ***MatsSorbentTrapRecord***.TrapId
3) BeginDateHour is before ***MatsSorbentTrapRecord***.EndDateHour
4) EndDateHour is after ***MatsSorbentTrapRecord***.BeginDateHour

If found,

Set ***MatsSorbentTrapEvaluationNeeded*** to false.
Return result A.

Results:

Result	Response	Severity
A	For [key], you reported sorbent traps with overlapping sampling periods.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Sorbent Trap Overlap Evaluation

Check Code: MATSTRP-7

Check Name: Initialize MATS Sorbent Trap Parameters

Related Former Checks:

Applicability:

Description: Initialize MATS Sampling Train Data

Specifications:

Set ***MatsSorbentTrapValidExists*** to false.
Set ***MatsSorbentTrapSamplingTrainList*** to null.
Set ***MatsSamplingTrainProblemComponentExists*** to false.

For ***MatsSorbentTrapRecord***:

Set *SorbentTrapInformation* record with:

1) SorbentTrapValidExists set to true.
2) IsBorderTrap set to (***MatsSorbentTrapRecord***.BorderTrapInd is equal to 1).
3) IsSupplementalData set to (***MatsSorbentTrapRecord***.SupplementalDataInd is equal to 1).
4) SorbentTrapId set to ***MatsSorbentTrapRecord***.TrapId
5) SorbentTrapBeginDateHour set to ***MatsSorbentTrapRecord***.BeginDateHour
6) SorbentTrapEndDateHour set to ***MatsSorbentTrapRecord***.EndDateHour
7) SorventTrapModcCd set to ***MatsSorbentTrapRecord***.ModcCd
8) SamplingTrainProblemComponentExists set to false.
9) SamplingTrainList with a record containing the following fields:
    a) HgConcentration as a decimal
    b) TrainQAStatusCode as a string
    c) ReferenceSFSRRatio as an integer
    d) TotalSFSRRatioCount as an integer
    e) DeviatedSFSRRatioCount as an integer
    f) SamplingTrainValid as a boolean
10) OperatingDateList set to an empty list of dates.

Set ***MatsSorbentTrapDictionary*** to *SorbentTrapInformation*, where the TrapId key is equal ***MatsSorbentTrapRecord***.TrapId.
Append *SorbentTrapInformation* to ***MatsSorbentTrapListByLocationArray*** element for ***CurrentMonitorPlanLocationPosition***.

Set ***MatsSorbentTrapValidExists*** to ***MatsSorbentTrapDictionary***.SorbentTrapValidExists where ***MatsSorbentTrapDictionary*** key is equal to ***MatsSorbentTrapRecord***.TrapId.
Set ***MatsSorbentTrapSamplingTrainList*** to ***MatsSorbentTrapDictionary***.SamplingTrainList where ***MatsSorbentTrapDictionary*** key is equal to ***MatsSorbentTrapRecord***.TrapId.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Sorbent Trap First Hour Initialization

Check Code: MATSTRP-8

Check Name: Monitoring System Check

Related Former Checks:

Applicability:

Description: Ensure that Monitoring System exists for Monitoring System ID, and that the Monitoring System type is “ST”.

Specifications:

If the ***MatsSorbentTrapRecord***.MonitoringSystemID is null,

Set ***MatsSorbentTrapValidExists*** to false.
Return result A.

Else if the ***MatsSorbentTrapRecord***.SystemTypeCode of the associated system is not equal to "ST",

Set ***MatsSorbentTrapValidExists*** to false.
Return result B.

Else if ***MatsSorbentTrapRecord***.SystemBeginDateHour is after ***MatsSorbentTrapRecord***.BeginDateHour, OR ***MatsSorbentTrapRecord***.SystemEndDateHour is NOT null and is before ***MatsSorbentTrapRecord***.EndDateHour,

Set ***MatsSorbentTrapValidExists*** to false.
Return result C.

Results:

Result	Response	Severity
A	For [key], you have not reported a value for [fieldname], which is required.	Critical Error Level 1
B	The SystemTypeCode in the monitoring plan is [system type]. This type of system does not report sorbent trap data.	Critical Error Level 1
C	The system reported for the sorbent trap does not span the period of the sorbent trap.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- MATS Sorbent Trap Evaluation

Check Code: MATSTRP-9

Check Name: Number and Validity of Sampling Trains

Related Former Checks:

Applicability:

Description: Check that two Sorbent Train Data Records are provided for each Sorbent Trap Data Record.

Specifications:

Set ***MatsSamplingTrainsValid*** to false,

If ***MatsSamplingTrainProblemComponentExists*** is false,

If number of entries in ***MatsSorbentTrapSamplingTrainList*** is not equal to 2,

Set ***MatsSorbentTrapValidExists*** to false.
Return result A.

Else if ***MatsSorbentTrapSamplingTrainList*** .SamplingTrainValid for one or both sampling train components is false,

Set ***MatsSorbentTrapValidExists*** to false.

Else

Set ***MatsSamplingTrainsValid*** to true

Results:

Result	Response	Severity
A	For [key], you did not report two sets of sorbent train records for the sorbent trap.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sorbent Trap Evaluation

Check Code: MATSTRP-10

Check Name: Sorbent Trap MODC Code is Valid

Related Former Checks:

Applicability:

Description: Check Sorbent Trap MODC Code Valid

Specifications:

Set ***MatsSorbentTrapMODCCodeValid*** to false

If ***MatsSorbentTrapRecord***.MODCCode is not equal to "01", "02", "32", "33", "34", "35", "43" or "44",

Set ***MatsSorbentTrapValidExists*** to false.
Return result A.

Else if ***MatsSamplingTrainsValid***,

If ***MatsSorbentTrapRecord***.MODCCode is equal to "01" or "02" or "43",

If ***MatsSorbentTrapSamplingTrainList***.TrainQAStatusCode for both sampling train components are equal to "PASSED",

Set ***MatsSorbentTrapMODCCodeValid*** equal to true.

Else
Set ***MatsSorbentTrapValidExists*** to false.
Return result B

Else if ***MatsSorbentTrapRecord***.MODCCode is equal to "32" or "44",

If ***MatsSorbentTrapSamplingTrainList***.TrainQAStatusCode are equal to "PASSED" for one sampling train component, AND "FAILED" or "LOST" for the other,

Set ***MatsSorbentTrapMODCCodeValid*** equal to true.

Else
Set ***MatsSorbentTrapValidExists*** to false.
Return result C

Else if the ***MatsSorbentTrapRecord***.MODCCode is equal to "33",

If ***MatsSorbentTrapSamplingTrainList***.TrainQAStatusCode for both sampling train components are equal to "UNCERTAIN",

Set ***MatsSorbentTrapMODCCodeValid*** equal to true.

Else
Set ***MatsSorbentTrapValidExists*** to false.
Return result D

Else if the ***MatsSorbentTrapRecord***.MODCCode is equal to "34",

If ***MatsSorbentTrapSamplingTrainList***.TrainQAStatusCode for both sampling train components are equal to "FAILED",

Set ***MatsSorbentTrapMODCCodeValid*** equal to true.

Else If ***MatsSorbentTrapSamplingTrainList***.TrainQAStatusCode for both sampling train components are equal to "UNCERTAIN",

Set ***MatsSorbentTrapMODCCodeValid*** equal to true.

Else if ***MatsSorbentTrapSamplingTrainList***.TrainQAStatusCode for one or both sampling train components is equal to "LOST", "EXPIRED" or "INC",

Set ***MatsSorbentTrapMODCCodeValid*** equal to true.

Else

Set ***MatsSorbentTrapValidExists*** to false.
Return result E

Else if the ***MatsSorbentTrapRecord***.MODCCode is equal to "35",

Set ***MatsSorbentTrapMODCCodeValid*** equal to true.

Results:

Result	Response	Severity
A	For [key] you reported a [value] which is not valid for [fieldname].	Critical Error Level 1
B	For [key], you reported a [fieldname] of [value] which is valid if the QA Status Codes of both trains is PASSED.	Critical Error Level 1
C	For [key], you reported a [fieldname] of [value] which is valid if the QA Status Code of one train is PASSED and the other FAILED.	Critical Error Level 1
D	For [key] you reported a [fieldname] of [value] which is valid if the QA Status Code of both trains is UNCERTAIN.	Critical Error Level 1
E	For [key], you reported a [fieldname] of [value] which is valid if the QA Status Code of both trains is FAILED or UNCERTAIN, or one or both trains is LOST, EXPIRED, or INC.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sorbent Trap Evaluation

Check Code: MATSTRP-11

Check Name: Paired Trap Agreement Validation and Re-calculation

Related Former Checks:

Applicability:

Description: Determine if the Paired Trap Agreement is Valid.

Specifications:

Set ***MatsSorbentTrapPairedTrapAgreementValid*** to false.
Set **MatsCalcTrapAbsoluteDifference** = null.
Set ***MatsCalcTrapPercentDifference*** = null.

If the ***MatsSorbentTrapMODCCodeValid*** is equal to true,

If ***MatsSorbentTrapRecord***.PairedTrapAgreement is null,

If ***MatsSorbentTrapRecord***.MODCCode is not "32", "34", "35", or "44",

Set ***MatsSorbentTrapValidExists*** to false.
Return result A.

Else if ***MatsSorbentTrapRecord***.AbsoluteDifferenceIndicator is not null,

Set ***MatsSorbentTrapValidExists*** to false.
Return result B.

Else
Set ***MatsSorbentTrapPairedTrapAgreementValid*** to true

Else

If ***MatsSorbentTrapRecord***.MODCCode is not "01", "02", "33", or "43",

Set ***MatsSorbentTrapValidExists*** to false.
Return result C.

Else if ***MatsSorbentTrapRecord***.PairedTrapAgreement is not rounded to two decimal places

Set ***MatsSorbentTrapValidExists*** to false.
Return result D.

Else if ***MatsSorbentTrapRecord***.AbsoluteDifferenceIndicator is equal to 0, OR
MatsSorbentTrapRecord.AbsoluteDifferenceIndicator is equal to 1,

Set **MatsCalcTrapAbsoluteDifference** = the absolute value of the difference between the ***MatsSorbentTrapSamplingTrainList***.HgConcentration for each train.

If the sum of the ***MatsSorbentTrapSamplingTrainList***.HgConcentration from each train is NOT equal to 0,
Set ***MatsCalcTrapPercentDifference*** = 100 * ***MatsCalcTrapAbsoluteDifference*** divided by the sum of the ***MatsSorbentTrapSamplingTrainList***.HgConcentration from each train.
Else
Set ***MatsCalcTrapPercentDifference*** = 0

Round ***MatsCalcTrapAbsoluteDifference*** to 2 decimal places.
Round ***MatsCalcTrapPercentDifference*** to 2 decimal place.

If ***MatsSorbentTrapRecord.***AbsoluteDifferenceIndicator is equal to 0,

If ***MatsSorbentTrapRecord***.PairedTrapAgreement does not equal ***MatsCalcTrapPercentDifference***,

Set ***MatsSorbentTrapValidExists*** to false.
Return result G.

Else if ***MatsSorbentTrapRecord***.PairedTrapAgreement is less than or equal to 10,

If ***MatsSorbentTrapRecord***.MODCCode is not equal to "01", "02" OR "43",

Set ***MatsSorbentTrapValidExists*** to false.
Return result H.

Else
Set ***MatsSorbentTrapPairedTrapAgreementValid*** to true.

Else if ***MatsSorbentTrapRecord***.PairedTrapAgreement is less than or equal to 20, and the ***MatsSorbentTrapRecord***.HgSystemConcentration is less than or equal to 1.0,

If ***MatsSorbentTrapRecord***.MODCCode is not equal to "01", "02" OR "43",
Set ***MatsSorbentTrapValidExists*** to false.
Return result I.

Else
Set ***MatsSorbentTrapPairedTrapAgreementValid*** to true.

Else

If ***MatsSorbentTrapRecord***.MODCCode is not equal to "33",

Set ***MatsSorbentTrapValidExists*** to false.
Return result J.

Else
Set ***MatsSorbentTrapPairedTrapAgreementValid*** to true.

Else // AbsoluteDifferenceIndicator is equal to 1

If ***MatsSorbentTrapRecord.***PairedTrapAgreement is less than or equal to 0.03,

If ***MatsSorbentTrapRecord***.PairedTrapAgreement does not equal ***MatsCalcTrapAbsoluteDifference***,

Set ***MatsSorbentTrapValidExists*** to false.
Return result E.

Else
Set ***MatsSorbentTrapPairedTrapAgreementValid*** to true.

Else
Set ***MatsSorbentTrapValidExists*** to false.
Return result F.

Else // AbsoluteDifferenceIndicator is null (not 0 or 1)

Set ***MatsSorbentTrapValidExists*** to false.
Return result K.

Results:

Result	Response	Severity
A	For [key], you did not report a [fieldname] value which is required if the sorbent trap system MODC Code is [value].	Critical Error Level 1
B	For [key], you reported a [fieldname2], but did not report a [fieldname].	Critical Error Level 1
C	For [key], you reported a [fieldname] value which is not reported if the sorbent trap system MODC Code is [value].	Critical Error Level 1
D	The [fieldname] value for [key] should be reported to two decimal places.	Critical Error Level 1
E	The [fieldname] is inconsistent with the value [value1] calculated from the reported [key] records used in the calculation.	Critical Error Level 1
F	For [key], you reported an Absolute Difference Indicator of 1 that can only be used if the absolute difference between the Hg concentrations of the paired traps is less than or equal to 0.03 ug/m3.	Critical Error Level 1
G	The [fieldname] is inconsistent with the value [value2] calculated from the reported [key] records used in the calculation.	Critical Error Level 1
H	For [key], you reported a passing Paired Trap Agreement that is not consistent with an MODC Code other than 01, 02 or 43.	Critical Error Level 1
I	For [key], you reported a passing Paired Trap Agreement that is not consistent with an MODC Code other than 01, 02 or 43.	Critical Error Level 1
J	For [key], you reported a failed Paired Trap Agreement that is not consistent with an MODC Code other than 33.	Critical Error Level 1
K	For [key], you did not report a [fieldname2] value which is required if the sorbent trap system MODC Code is [value].	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- MATS Sorbent Trap Evaluation

Check Code: MATSTRP-12

Check Name: Hg System Concentration Validation and Re-calculation

Related Former Checks:

Applicability:

Description: Determine if the Hg System Concentration is Valid.

Specifications:

Set ***MatsCalcHgSystemConcentration*** equal to null.

If the ***MatsSorbentTrapPairedTrapAgreementValid*** is equal to true,

If the ***MatsSorbentTrapRecord***.HgSystemConcentration is null,

If the ***MatsSorbentTrapRecord***.MODCCode is not "34" or "35",

Set ***MatsSorbentTrapValidExists*** to false.
Return result A.

Else
If the ***MatsSorbentTrapRecord***.MODCCode is not "01", "02", "32", "33", "43" or "44",

Set ***MatsSorbentTrapValidExists*** to false.
Return result B.

Else if the ***MatsSorbentTrapRecord***.HgSystemConcentration is not reported in scientific notation rounded to three significant figures, keeping one to the left of the decimal point (x.xx-E-x).

Set ***MatsSorbentTrapValidExists*** to false.
Return result C.

Else if the ***MatsSorbentTrapRecord***.HgSystemConcentration is 0,

Set ***MatsSorbentTrapValidExists*** to false.
Return result E

Else if any of the ***MatsSorbentTrapSamplingTrainList.***HgConcentration entries are 0,

Set ***MatsSorbentTrapValidExists*** to false.
Return result F

Else
If ***MatsSorbentTrapRecord***.MODCCode is equal to "32" or "44",
Set *HgConcentrationCalculation* = multiply 1.111 times the
MatsSorbentTrapSamplingTrainList.HgConcentration entry where
MatsSorbentTrapSamplingTrainList.TrainQAStatusCode is equal to "PASSED".

Else if ***MatsSorbentTrapDataRecord***.MODCCode is equal to "33" ,
Set *HgConcentrationCalculation* = the higher of the
MatsSorbentTrapSamplingTrainList.HgConcentration entries for the sampling train components.

Else //MODC "01", "02" or "43"
Set *HgConcentrationCalculation* = the sum of the
MatsSorbentTrapSamplingTrainList.HgConcentration for each train divided by two.

Set ***MatsCalcHgSystemConcentration*** to *HgConcentrationCalculation* in scientific notation with three significant digits, keeping one to the left of the dicmal point (x.xxEx).

If ***MatsSorbentTrapRecord***.HgSystemConcentration does not equal ***MatsCalcHgSystemConcentration***,

Set ***MatsSorbentTrapValidExists*** to false.
Return result D.

Results:

Result	Response	Severity
A	For [key], you did not report a [fieldname] value which is required if the sorbent trap system MODC Code is [value].	Critical Error Level 1
B	For [key], you reported a [fieldname] value which is not reported if the sorbent trap system MODC Code is [value].	Critical Error Level 1
C	The [fieldname] value in the [key] records is not reported in scientific notation rounded to three significant figures, with one digit to the left of the decimal point.	Critical Error Level 1
D	The [fieldname] is inconsistent with the value [value1] calculated from the reported [key] records used in the calculation.	Critical Error Level 1
E	For [key], you reported a 0 for the Hg Concentration at the Sorbent Trap. When the measured Hg concentration is less than the detection limit, the Method Detection Limit should be reported at the Sorbent Trap and one or both Sampling Trains, as per 63.10007(e)(1).	Critical Error Level 1
F	For [key], you reported a 0 for the Hg Concentration at one or both Sampling Trains. When the measured Hg concentration is less than the detection limit, the Method Detection Limit should be reported instead of zero, as per 63.10007(e)(1).	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- MATS Sorbent Trap Evaluation

Check Code: MATSTRP-13

Check Name: Update Sorbent Trap Parameters

Related Former Checks:

Applicability:

Description:

Specifications:

For ***SorbentTrapDictionary*** entry where the key is equal to ***MatsSorbentTrapRecord***.TrapId, set:

1) SorbentTrapValidExists set to ***MatsSorbentTrapValidExists*** .
2) SamplingTrainProblemComponentExists set to ***MatsSamplingTrainProblemComponentExists***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- MATS Sorbent Trap Evaluation

Check Code: MATSTRP-14

Check Name: Number of Unit Operating Days

Related Former Checks:

Applicability:

Description: Check the Number of Unit Operating Days During Sampling Period.

Specifications:

If ***MatsSorbentTrapRecord***.ModcCode is NOT equal to "34",

If ***MatsSorbentTrapDictionary***.OperatingDateList where ***MatsSorbentTrapDictionary*** key is equal to ***MatsSorbentTrapRecord***.TrapId contains more than 15 dates,

Return result A.

Results:

Result	Response	Severity
A	You reported a sorbent trap sampling period longer than the maximum 15 operating days.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report ------ MATS Sorbent Trap Last Hour Evaluation

Check Code: MATSTRP-15

Check Name: Ensure that Active Methods Span the Sorbent Trap Period.

Related Former Checks:

Applicability: General Check

Description: This check ensures that active ST or CEMST methods exists and span the period of the sorbent trap.

Specifications:

Locate ***MethodRecords*** for the location where:
1) MethodCode is equal to "ST" or "CEMST".
2) BeginDateHour is less than or equal to ***MatsSorbentTrapRecord***.EndDateHour.
3) EndDateHour is null OR greater than or equal to ***MatsSorbentTrapRecord***.BeginDateHour.

If not found,
return result A.

Else if BeginDateHour and EndDateHour of the retrieved ***MethodRecords*** records do not span the entire period of the ***MatsSorbentTrapRecord*** BeginDateHour and EndDateHour,
return result B.

Results:

Result	Response	Severity
A	You reported sorbent trap [KEY], but no sorbent trap methods are active during the period the trap is active.	Critical Error Level 1
B	You reported sorbent trap [KEY], but the sorbent trap methods are not active for the entire period the trap is active.	Critical Error Level 1

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- MATS Sorbent Trap Evaluation

Check Category:

NSPS4T Summary, Compliance Period and Annual Checks

Check Code: NSPS4T-1

Check Name: NSPS4T Summary: Validate Record Count

Related Former Checks:

Applicability:

Description: Ensures that the correct number of NSPS4T Summary rows exists, based on whether NSPS4T is an active program.

Specifications:

Set ***Nsps4tCurrentSummaryRecord*** to null.

Locate *CurrentSummaryRecords* in ***Nsps4tSummaryRecords*** where MonitorLocationKey is equal to ***CurrentMonitorPlanLocationRecord***.MonitorLoctionKey.

If the first two characters of ***CurrentMonitorPlanLocationRecord***.LocationName do not equal to "CS", "MS", "CP" or "MP",

Locate *CurrentLocationProgramRecord* in ***EmLocationProgramRecords*** where:

1) ProgramCode is equal to "NSPS4T".
2) ClassCode is equal to "A".
3) UnitMonitorCertBeginDate is on or before the ***CurrentReportingPeriodEndDate***.
4) EndDate is null, OR is on or after ***CurrentReportingPeriodBeginDate***.

If *CurrentLocationProgramRecord* does not exist,

If the count of *CurrentSummaryRecords* is greater than 0,

Return result A.

Else

If the count of *CurrentSummaryRecords* is greater than 1,

Return result B.

Else

Set ***Nsps4tCurrentSummaryRecord*** to the single record in *CurrentSummaryRecords*.

Else

If the count of *CurrentSummaryRecords* is greater than 0,

Return result C.

Results:

Result	Response	Severity
A	You have reported data for the NSPS4T program, but based on information in the monitoring plan, this unit is not NSPS4T affected.	Critical Error Level 1
B	You have reported more than one set of NSPS4TSummary data for this location.	Critical Error Level 1
C	You have reported NSPS4T summary data for a stack or pipe, but the data is only allowed for a unit.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report NSPS4T Summary Data Evaluation

Check Code: NSPS4T-2

Check Name: NSPS4T Compliance Period: Validate Record Count

Related Former Checks:

Applicability:

Description: Ensures that an allowed number of NSPS4T Compliance Period rows exists.

Specifications:

Set ***Nsps4tCurrentCompliancePeriod1Record*** to null.
Set ***Nsps4tCurrentCompliancePeriod2Record*** to null.
Set ***Nsps4tCurrentCompliancePeriod3Record*** to null.

If ***Nsps4tCurrentSummaryRecord*** is NOT null,

Locate *CurrentCompliancePeriodRecords* in ***Nsps4tCompliancePeriodRecords*** where Nsps4tSummaryKey is equal to ***Nsps4tCurrentSummaryRecord***.Nsps4tSummaryKey.

If ***Nsps4tCurrentSummaryRecord***.NoCompliancePeriodEndedIndicator is equal to 1,

If the count of *CurrentCompliancePeriodRecords* is greater than 0,

Return result A.

Else
If the count of *CurrentCompliancePeriodRecords* is greater than 3,

Return result B.

Else if the count of *CurrentCompliancePeriodRecords* is equal to 0,

Return result C.

Else
Set ***Nsps4tCurrentCompliancePeriod1Record*** to the first record in *CurrentCompliancePeriodRecords*.
Set ***Nsps4tCurrentCompliancePeriod2Record*** to the second record in *CurrentCompliancePeriodRecords*, if it exists
Set ***Nsps4tCurrentCompliancePeriod3Record*** to the third record in *CurrentCompliancePeriodRecords*, if it exists.

Results:

Result	Response	Severity
A	You have reported NoCompliancePeriodEndedIndicator of 1, but have also reported compliance period data. This is incorrect.	Critical Error Level 1
B	You have reported compliance period data for more than 3 periods. This is incorrect.	Critical Error Level 1
C	You have reported NoCompliancePeriodEndedIndicator of 0 but have not reported compliance period data.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report NSPS4T Summary Data Evaluation

Check Code: NSPS4T-3

Check Name: NSPS4T Annual (4th Quarter): Validate Record Count

Related Former Checks:

Applicability:

Description: Ensures that the correct number of NSPS4T Annual rows exists.

Specifications:

Set ***Nsps4tCurrentAnnualRecord*** to null.

If ***Nsps4tCurrentSummaryRecord*** is not null,

Locate *CurrentAnnualRecords* in ***Nsps4tAnnualRecords*** where Nsps4tSummaryKey is equal to ***Nsps4tCurrentSummaryRecord***.Nsps4tSummaryKey.

If ***CurrentReportingPeriodQuarter*** is not equal to 4,

If the count of *CurrentAnnualRecords* is greater than 0,

Return result A.

Else
If the count of *CurrentAnnualRecords* is greater than 1,

Return result B.

Else if the count of *CurrentAnnualRecords* is equal to 0,

Return result C.

Else
Set ***Nsps4tCurrentAnnualRecord*** to the single record in *CurrentAnnualRecords*.

Results:

Result	Response	Severity
A	You have reported NSPS4T Fourth Quarter data. This data should only be reported in fourth quarter files.	Critical Error Level 1
B	You have reported more than one NSPS4T Fourth Quarter record. You should only report one record per location.	Critical Error Level 1
C	You have not reported NSPS4T Fourth Quarter data. A fourth quarter file should contain this data.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report NSPS4T Summary Data Evaluation

Check Code: NSPS4T-4

Check Name: NSPS4T Summary CO2 Emissions Standard vs. Electrical Load Code

Related Former Checks:

Applicability:

Description: Ensures that the NSPS4T Summary's CO2 Emissions Standard and Electrical Load are consistent with each other.

Specifications:

If ***Nsps4tCurrentSummaryRecord*** is not null,

If ***Nsps4tCurrentSummaryRecord***.Co2EmissionStandardElectricalLoadCode is not null, AND does not equal ***Nsps4tCurrentSummaryRecord***.ElectricalLoadCode,

Return result A.

Results:

Result	Response	Severity
A	You reported an Electrical Load that is not consistent with the reported CO2 Emission Standard.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report NSPS4T Summary Data Evaluation

Check Code: NSPS4T-5

Check Name: NSPS4T Summary CO2 Emissions Standard vs. Compliance Period CO2 Emission Rate UOM

Related Former Checks:

Applicability:

Description: Ensures that the NSPS4T Summary's CO2 Emissions Standard and each reported Compliance Period CO2 Emission Rate UOM are consistent with each other.

Specifications:

Set ***Nsps4tInvalidCo2EmissionRateUomList*** to "".

If ***Nsps4tCurrentSummaryRecord*** is not null,

If ***Nsps4tCurrentSummaryRecord***.Co2EmissionStandardRateUomCode is not null,

If ***Nsps4tCurrentCompliancePeriod1Record*** is not null,

If ***Nsps4tCurrentCompliancePeriod1Record***.CO2EmissionRateUOMCode is not null, AND is not equal to ***Nsps4tCurrentSummaryRecord***.Co2EmissionStandardRateUomCode,

Append ***Nsps4tCurrentCompliancePeriod1Record***.CO2EmissionRateUOMLabel to ***Nsps4tInvalidCo2EmissionRateUomList***.

If ***Nsps4tCurrentCompliancePeriod2Record*** is not null,

If ***Nsps4tCurrentCompliancePeriod2Record***.CO2EmissionRateUOMCode is not null, AND is not equal to ***Nsps4tCurrentSummaryRecord***.Co2EmissionStandardRateUomCode,

Append ***Nsps4tCurrentCompliancePeriod2Record***.CO2EmissionRateUOMLabel to ***Nsps4tInvalidCo2EmissionRateUomList***.

If ***Nsps4tCurrentCompliancePeriod3Record*** is not null,

If ***Nsps4tCurrentCompliancePeriod3Record***.CO2EmissionRateUOMCode is not null, AND is not equal to ***Nsps4tCurrentSummaryRecord***.Co2EmissionStandardRateUomCode,

Append ***Nsps4tCurrentCompliancePeriod3Record***.CO2EmissionRateUOMLabel to ***Nsps4tInvalidCo2EmissionRateUomList***.

If ***Nsps4tInvalidCo2EmissionRateUomList*** does not equal "",

Return result A.

Else if ***Nsps4tCurrentSummaryRecord***.Co2EmissionStandardCode is equal to "MODUS",

If ***Nsps4tCurrentCompliancePeriod1Record*** is not null,

If ***Nsps4tCurrentCompliancePeriod2Record*** is not null, AND
Nsps4tCurrentCompliancePeriod2Record.CO2EmissionRateUOMCode is not equal to ***Nsps4tCurrentCompliancePeriod1Record***.CO2EmissionRateUOMCode,
OR
Nsps4tCurrentCompliancePeriod3Record is not null, AND
Nsps4tCurrentCompliancePeriod3Record.CO2EmissionRateUOMCode is not equal to ***Nsps4tCurrentCompliancePeriod1Record***.CO2EmissionRateUOMCode,

Return result B.

Results:

Result	Response	Severity
A	You have used different units of measure in your reporting of CO2 emission rate codes. You should use the same CO2EmissionRateUOMCode in all instances.	Informational Message
B	The reported NSPS4T CO2 emission rate UOM values in your reported compliance period records do not match.	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report NSPS4T Summary Data Evaluation

Check Category:

RATA Status

Check Code: RATSTAT-1

Check Name: Check Low Sulfur and FLOW Exemptions

Related Former Checks:

Applicability: CEM Check

Description: This check determines if an SO2 system is exempt for RATA Status purposes or determines the maximum number of levels applicable to a FLOW system for determination of RATA Status.

Specifications:

Set ***CurrentRATAStatus*** = null.
Set ***OverrideRATABAF*** = null.
Set ***MaxLevelCount*** = null.
Set ***FlowRATAExemption*** = false.

if (***QaStatusSystemTypeCode*** begins with "SO2")

Locate a record in ***TestExtensionExemptionRecords*** for the location where the SystemID is equal to the ***QaStatusSystemId***, the reporting period is the Current Reporting Period, AND the ExtensionExemptionCode is equal to "LOWSYTD"

if (***TestExtensionExemptionRecords*** is found)

Set ***CurrentRATAStatus*** = "IC-Exempt"

else

Locate a record in ***MonitorQualificationRecordsByHour*** for the hour and where MonitoringLocationId = ***CurrentMonitorLocationId***
and the QualificationTypeCode is equal to "LOWSULF".

if (***MonitorQualificationRecordsByHour*** is found)

Set ***CurrentRATAStatus*** = "IC-Exempt".

else if (***QaStatusSystemTypeCode*** = "FLOW")

Set *PeakingBypass* = false.

if (***CurrentEntityType*** is equal to "CS" or "MS")

Locate the record in ***LocationAttributeRecordsByHourLocation*** for the hour and location.

if (***LocationAttributeRecords***.BypassInd == 1)

Set *PeakingBypass* = true.

else

Set *PeakingBypass* = true.

For each record in ***UnitStackConfigurationRecordsByHourLocation*** for the hour and stack location

Locate a record in ***MonitorQualificationRecordsByHour*** for the hour where MonitoringLocationId = the unit location in the ***UnitStackConfigurationRecordsByHourLocation*** record and QualificationTypeCode is equal to "PK" or "SK".

if (***MonitorQualificationRecordsByHour*** is NOT found)

Set *PeakingBypass* = false.
Exit for.

else if (***CurrentUnitisPeaking***)

Set *PeakingBypass* = true.

if (*PeakingBypass)*

Set ***MaxLevelCount*** = 1.

else

Locate a record in ***MonitorQualificationRecordsByHour*** for the hour where the MonitoringLocationId = ***CurrentMonitorLocationId*** and QualificationTypeCode is equal to "PRATA1"

if (***MonitorQualificationRecordsByHour*** is found)

Set ***MaxLevelCount*** = 1.

else

Locate a record in ***MonitorQualificationRecordsByHour*** for the hour where MonitoringLocationId = ***CurrentMonitorLocationId*** and the QualificationTypeCode is equal to "PRATA2"

if (***MonitorQualificationRecordsByHour*** is found)

Set ***MaxLevelCount*** = 2.

else

Set ***MaxLevelCount*** = 3.
Append ***QaStatusSystemId*** to ***FLOWSystemIDArray*** for the location.

Locate a record in ***TestExtensionExemptionRecords*** for the location where the SystemID is equal to the ***QaStatusSystemId***, the ComponentID is equal to the the ***QaStatusComponentId*** the reporting period is the Current Reporting Period, AND the ExtensionExemptionCode is equal to "FLOWEXP"

if (***TestExtensionExemptionRecords*** is found)
Set ***FlowRATAExemption*** = true.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2/O2 RATA Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- H2O RATA Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- H2OM RATA Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- Hg RATA Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOX RATA Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- NOXC RATA Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB NOX RATA Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 RATA Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report --------- Stack Flow RATA Status Evaluation

Check Code: RATSTAT-2

Check Name: Locate Most Recent Prior RATA Test

Related Former Checks:

Applicability: CEM Check

Description: Determines if there is an applicable prior RATA test.

Specifications:

Set ***PriorRATARecord*** = null.
Set ***InvalidRATARecord*** = null.
Set ***ApplicableSystemIDList*** = null.

If ***Flow RATA Exemption*** is true,

Locate all ***MonitorSystemComponentRecordsforHourandLocation*** for the location and hour where the ComponentID is equal to ***QaStatusComponentId***.

For each record found,
Append ***MonitorSystemComponentRecordsforHourandLocation***.SystemID to the ***ApplicableSystemIDList***.

else
Append ***QaStatusSystemId*** to the ***ApplicableSystemIDList***.

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is in the ***ApplicableSystemIDList*** and the TestResultCode is not equal to "INVALID" and the EndDate/Hour is prior to the ***CurrentDateHour***

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***PriorRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

if (***CurrentRATAStatus*** is null)

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is in the ***ApplicableSystemIDList*** and the EndDate/Hour is prior to the ***CurrentDateHour*** and the EndDate/Hour is greater than the ***PriorRATARecord***.EndDate/Hour and the TestResultCode is equal to "INVALID".

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***InvalidRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

else

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is in the ***ApplicableSystemIDList*** and the TestResultCode is equal to "INVALID" and the EndDate/Hour is prior to the ***CurrentDateHour***

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***InvalidRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2/O2 RATA Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- H2O RATA Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- H2OM RATA Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- Hg RATA Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOX RATA Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- NOXC RATA Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB NOX RATA Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 RATA Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report --------- Stack Flow RATA Status Evaluation

Check Code: RATSTAT-3

Check Name: Locate Most Recent Prior Event

Related Former Checks:

Applicability: CEM Check

Description: Determines if there is an applicable prior event.

Specifications:

Set ***PriorRATAEventRecord*** = null.

If (***CurrentRATAStatus*** is null)

Locate the most recent record in ***QACertificationEventRecords*** where the SystemID is in the ***ApplicableSystemIDList*** and RATARequired is equal to "Y" and the QACertEventDate is either:

a) prior to the ***CurrentDateHour*** OR
b) equal to both the ***CurrentDateHour*** and the ConditionalBeginDate/Hour;

AND either:

a) ***PriorRATARecord*** is null OR
b) QACertEventDate/Hour is after the ***PriorRATARecord***.EndDate/Hour OR
c) QACertEventDate/Hour is equal to the ***PriorRATARecord***.EndDate/Hour AND (TestCompletionDate is null or the TestCompletionDate/Hour is after the ***PriorRATARecord***.EndDate/Hour)

AND either:

a) ***Annual Reporting Requirement*** is equal to true OR
b) QACertEventDate/Hour is on or after April 1 of the year of ***CurrentDateHour***

AND either:

a) ***QaStatusSystemTypeCode*** NOT is set (HCL, HF, HG, ST) OR
b) QACertEventCode is in set (101, 110, 125, 130)

if (***QACertificationEventRecords*** is found)

Set ***PriorRATAEventRecord*** = the found record in ***QACertificationEventRecords***.

if (***PriorRATAEventRecord*** is null)

if (***PriorRATARecord*** is null)

Set ***CurrentRATAStatus*** = "OOC-No Prior Test or Event"

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = 1.0

else if (***InvalidRATARecord*** is not null AND ***PriorRATAEventRecord***.QACertEventDate/Hour is after ***InvalidRATARecord***.EndDate/Hour)

Locate the earliest record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***QaStatusSystemId*** and the EndDate/Hour is after the ***PriorRATAEventRecord***.QACertEventDate/Hour and the EndDate/Hour is prior to ***CurrentDateHour*** and the TestResult is equal to "INVALID"

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***InvalidRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

else

Set ***InvalidRATARecord*** = null.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2/O2 RATA Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- H2O RATA Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- H2OM RATA Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- Hg RATA Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOX RATA Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- NOXC RATA Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB NOX RATA Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 RATA Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report --------- Stack Flow RATA Status Evaluation

Check Code: RATSTAT-4

Check Name: Check RATA Result

Related Former Checks:

Applicability: CEM Check

Description: Checks the result of the prior Rata test.

Specifications:

if (***CurrentRATAStatus*** is null and ***PriorRATARecord*** is not null and ***PriorRATAEventRecord*** is null)

Set ***EvaluateMultiLevelRATA*** = true.

if (***PriorRATARecord***.QANeedsEvaluationFlag = "Y")

Set ***CurrentRATAStatus*** = "Prior Test Not Yet Evaluated".

else if (***PriorRATARecord***.TestResultCode = null or ***PriorRATARecord***.TestResultCode = "FAILED" or ***PriorRATARecord***.TestResultCode = "ABORTED")

Locate the most recent record in ***QACertificationEventRecords*** where the SystemID is in the ***ApplicableSystemIDList*** and RATARequired is equal to "Y" and the ConditionalBeginDate/Hour is:

a) on or prior to the ***CurrentDateHour*** AND
b) on or after the ***PriorRATARecord***.EndDate/Hour; AND
c) ***Annual Reporting Requirement*** is equal to true OR QACertEventDate/Hour is on or after April 1 of the year of the ***CurrentDateHour***.

if (***QACertificationEventRecords*** is found)

Set ***PriorRATAEventRecord*** = found record in ***QACertificationEventRecords***.

elseif (***PriorRATARecord***.TestResultCode = null)

Set ***CurrentRATAStatus*** = "OOC-Prior Test Has Critical Errors".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

else if (***PriorRATARecord***.TestResultCode = "FAILED")

Set ***CurrentRATAStatus*** = "OOC-Prior Test Failed".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

else if (***PriorRATARecord***.TestResultCode = "ABORTED")

Set ***CurrentRATAStatus*** = "OOC-Prior Test Aborted".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2/O2 RATA Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- H2O RATA Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- H2OM RATA Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- Hg RATA Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOX RATA Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- NOXC RATA Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB NOX RATA Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 RATA Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report --------- Stack Flow RATA Status Evaluation

Check Code: RATSTAT-5

Check Name: Determine Event Conditional Status

Related Former Checks:

Applicability: CEM Check

Description: If a QA Cert Event was found that affects this MHV record, evaluate the conditional status.

Specifications:

Set ***SubsequentRATARecord*** = null.
Set ***RATAMissingOpDataInfo*** = null.
Set ***RATA Event Operating Level Count*** = null.

if (***CurrentRATAStatus*** is null and ***PriorRATAEventRecord*** is not null)

if (***PriorRATAEventRecord***.ConditionalDataBeginDate/Hour is null or ***CurrentDateHour*** is prior to the ***PriorRATAEventRecord***.ConditionalDataBeginDate/Hour)

Set ***CurrentRATAStatus*** = "OOC-Event".
Set ***OverrideRATABAF*** = 1.0.

else

Locate the earliest record in ***RATATestRecordsByLocationForQAStatus*** where the SystemID is equal to the ***PriorRATAEventRecord***.SystemID, the TestResult is not equal to "INVALID" and the EndDate/Hour is on or after the ***PriorRATAEventRecord***.ConditionalDataBeginDate/Hour.

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***SubsequentRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

if (***RATATestRecordsByLocationForQAStatus***.QANeedsEvaluationFlag = "Y")

Set ***CurrentRATAStatus*** = "Recertification Test Not Yet Evaluated".

else if (***RATATestRecordsByLocationForQAStatus***.TestResultCode is null)

Set ***CurrentRATAStatus*** = "OOC-Recertification Test Has Critical Errors".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = ***SubsequentRATARecord***.OverallBiasAdjustmentFactor.

else if (***RATATestRecordsByLocationForQAStatus***.TestResultCode = "FAILED")

Set ***CurrentRATAStatus*** = "OOC-Recertification Test Failed".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = ***SubsequentRATARecord***.OverallBiasAdjustmentFactor.

else if (***RATATestRecordsByLocationForQAStatus***.TestResultCode = "ABORTED")

Set ***CurrentRATAStatus*** = "OOC-Recertification Test Aborted".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = ***SubsequentRATARecord***.OverallBiasAdjustmentFactor.

else
if (***PriorRATARecord*** is null)

RequiredLevelCount = ***MaxLevelCount***

else (if ***PriorRATAEventRecord***.RATA3Required is equal to "Y")
RequiredLevelCount = 3

else (if ***PriorRATAEventRecord***.RATA2Required is equal to "Y")
RequiredLevelCount = 2

else
RequiredLevelCount = 1

If (number of levels in ***RATATestRecordsByLocationForQAStatus***.OpLevelCodeList is less than *RequiredLevelCount)*

Set ***CurrentRATAStatus*** = "OOC-Incomplete Recertification".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = ***SubsequentRATARecord***.OverallBiasAdjustmentFactor.

else
Set ***RATA Event Operating Level Count*** to the *RequiredLevelCount*.

If (***InvalidRATARecord*** is null)

Locate the earliest record in ***RATATestRecordsByLocationForQAStatus*** where the SystemID is equal to the ***PriorRATAEventRecord***.SystemID**,** the TestResult is equal to "INVALID" and the EndDate/Hour is on or after the ***PriorRATAEventRecord***.ConditionalDataBeginDate/Hour and is before the EndDate/EndHour of the ***RATATestRecordsByLocationForQAStatus*** record retrieved above.

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***InvalidRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

else
if (***PriorRATAEventRecord***.RATA3Required is equal to "Y")
Set ***RATA Event Operating Level Count*** to 3.

else (if ***PriorRATAEventRecord***.RATA2Required is equal to "Y")
Set ***RATA Event Operating Level Count*** to 2.

else
Set ***RATA Event Operating Level Count*** to 1.

if (***CurrentRATAStatus*** is null AND ***Annual Reporting Requirement*** == false)

If (***SubsequentRATARecord*** is not null and ***SubsequentRATARecord***.EndDate/Hour is greater than October 30th of the year of the ***CurrentDateHour***) OR (***SubsequentRATARecord*** is null and the ***CurrentDateHour*** is in the 3rd quarter))

Set ***CurrentRATAStatus*** = "OOC-Conditional Period Expired".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = 1.0.

if (***CurrentRATAStatus*** is null)

if (***PriorRATAEventRecord***.RATACertEvent == "Y") and (***PriorRATAEventRecord***.SystemTypeCode is <>

"HF")

if ((***CurrentMhvParameter*** <> "FLOW" and ***PriorRATAEventRecord***.EventCode = 125) or (***CurrentMhvParameter*** == "FLOW" and ***PriorRATAEventRecord***.EventCode = 305))

if (the associated BeginDate of the system in the ***PriorRATAEventRecord*** is null)

Set ***CurrentRATAStatus*** = "Invalid Monitor System"

else

If (the associated SystemTypeCode of the system in the ***PriorRATAEventRecord*** == "SO2")

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where the ProgramCode is in ***ProgramRequiresSo2SystemCertificationList*** and the UnitMonitorCertBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorRATAEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)
Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the ProgramCode is in ***ProgramRequiresSo2SystemCertificationList*** and the EmissionsRecordingBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorRATAEventRecord***.

else if (the associated SystemTypeCode of the system in the ***PriorRATAEventRecord*** == "NOX")

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where the ProgramCode is in ***ProgramRequiresNoxSystemCertificationList*** and the UnitMonitorCertBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorRATAEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)
Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the ProgramCode is in ***ProgramRequiresNoxSystemCertificationList*** and the EmissionsRecordingBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorRATAEventRecord***.

else if (the associated SystemTypeCode of the system in the ***PriorRATAEventRecord*** == "NOXC")

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where the ProgramCode is in set ***ProgramRequiresNoxcSystemCertificationList*** and the UnitMonitorCertBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorRATAEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)
Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the ProgramCode is in set ***ProgramRequiresNoxcSystemCertificationList*** and the EmissionsRecordingBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorRATAEventRecord***.

else if (the associated SystemTypeCode of the system in the ***PriorRATAEventRecord*** in set ("HCL, HG, ST"))

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where the ProgramCode is in set {MATS} and the UnitMonitorCertBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorRATAEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)
Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the ProgramCode is in set {MATS} and the EmissionsRecordingBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorRATAEventRecord***.

else

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest UnitMonitorCertBeginDate where the UnitMonitorCertBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorRATAEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)
Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the EmissionsRecordingBeginDate is ON OR BEFORE the associated BeginDate of the system in the ***PriorRATAEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is not found)

Set ***CurrentRATAStatus*** = "Missing Program".

else if (***LocationProgramRecordsByHourLocation***UnitMonitorCertDeadline is not null)

if (CurrentDate is prior to the ***LocationProgramRecordsByHourLocation***.UnitMonitorCertDeadline)

Set ***CurrentRATAStatus*** = "IC-Conditional".

else
Set ***CurrentRATAStatus*** = "OOC-Conditional Period Expired".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = 1.0.

else

if (CurrentDate is prior to the ***LocationProgramRecordsByHourLocation***.UnitMonitorCertBeginDate + 180 days)
Set ***CurrentRATAStatus*** = "IC-Conditional".

else
Set ***CurrentRATAStatus*** = "OOC-Conditional Period Expired".

if (***CurrentMhvParameter*** == "FLOW")

Set ***OverrideRATABAF*** = 1.0.

else

If (the number of calendar days ON OR AFTER the ***PriorRATAEventRecord***.QACertEventDate and ON OR BEFORE the ***CurrentDateHour*** > 180)

Set ***CurrentRATAStatus*** = "OOC-Conditional Period Expired".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = 1.0.

else if (the quarter of the ***PriorRATAEventRecord***.QACertEventDate is equal to the quarter of the ***CurrentDateHour***)

If (the number of calendar days ON OR AFTER the ***PriorRATAEventRecord***.QACertEventDate and ON OR BEFORE the ***CurrentDateHour*** > 90)

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusSystemTypeCode*** is "NOX")

If (Days in ***QaCertEventSuppDataDictionaryArray*** for the current location and QA Cert Event Date where QaCertEventKey is equal to ***PriorRATAEventRecord***.QaCertEventKey > 90)

Set ***CurrentRATAStatus*** = "OOC-Conditional Period Expired".

Else
Set ***CurrentRATAStatus*** = "IC-Conditional".

Else

If (***Rpt Period Op Hours Accumulator Array*** for the location == -1)
Set ***CurrentRATAStatus*** = "Invalid Op Data".

else if (the number of calendar days ON OR AFTER the ***PriorRATAEventRecord***.QACertEventDate and ON OR BEFORE the ***CurrentDateHour*** is equal to ***Rpt Period Op Days Accumulator Array*** for the location)

Set ***CurrentRATAStatus*** = "OOC-Conditional Period Expired".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = 1.0.

else
Set ***CurrentRATAStatus*** = "IC-Conditional".

else
Set ***CurrentRATAStatus*** = "IC-Conditional".

else if (***PriorRATAEventRecord***.MinOpDaysPriorQuarter is null)
Set ***PriorRATAEventRecord***.MinOpDaysPriorQuarter = 0
Set ***PriorRATAEventRecord***.MaxOpDaysPriorQuarter = 0

For each quarter beginning with the quarter of the ***PriorRATAEventRecord***.QACertEventDate and continuing through the quarter

BEFORE the ***CurrentDateHour***:

if (***EarliestLocationReportDate*** <= the last day of the quarter being checked)

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusSystemTypeCode*** is “NOX”)

Locate a record in ***SystemOperatingSuppDataRecordsByLocation*** where:

1) SystemId is equal to ***QaStatusSystemId***.
2) Year is equal to the year being checked.
3) Quarter is equal to the quarter being checked.
4) OpSuppDataTypeCode = "OP".

If (***SystemOperatingSuppDataRecordsByLocation*** is found)
Set *OperatingDayCount* = ***SystemOperatingSuppDataRecordsByLocation***.Days.
Else
Set *OperatingDayCount* = null.

Else
Locate the record in ***OperatingSuppDataRecordsbyLocation*** where the OpTypeCode is equal to "OPDAYS" and the reporting period is equal to the quarter being checked.

If (***OperatingSuppDataRecordsbyLocation*** is found)
Set *OperatingDayCount* = ***OperatingSuppDataRecordsByLocation***.OpValue.
Else
Set *OperatingDayCount* = null.

if (*OperatingDayCount* is null)

Locate the record in ***ReportingFrequencyByLocation*** where CalendarYear/Quarter are equal to the quarter being checked.

if found, and (the quarter being checked is 2 or 3, or ***ReportingFrequencyByLocation***.ReportingFrequencyCode is equal to "Q"),

Set ***PriorRATAEventRecord***.MinOpDaysPriorQuarter = -1
Set ***RATAMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
exit for.

else if (the quarter being checked is the quarter of the ***PriorRATAEventRecord***.QACertEventDate)

supplementalCount = null.

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusSystemTypeCode*** is “NOX” AND ***PriorRATAEventRecord***.QaCertEventDateSystemSuppDataExists is true)

supplementalCount = ***PriorRATAEventRecord***.QaCertEventSystemOpDaysCount.

If (*supplementalCount* is null AND ***PriorRATAEventRecord***.QaCertEventDateSuppDataExists is true)

supplementalCount = ***PriorRATAEventRecord***.QaCertEventOpDaysCount.

If (*supplementalCount* is NOT null)

Set ***PriorRATAEventRecord***.MinOpDaysPriorQuarter = ***PriorRATAEventRecord***.MinOpDaysPriorQuarter + *supplementalCount*.
Set ***PriorRATAEventRecord***.MaxOpDaysPriorQuarter = ***PriorRATAEventRecord***.MaxOpDaysPriorQuarter + *supplementalCount*.

Else

If (*OperatingDayCount* MINUS the number of calendar days in the quarter being checked that are PRIOR to the ***PriorRATAEventRecord***.QACertEventDate > 0)

Set ***PriorRATAEventRecord***.MinOpDaysPriorQuarter = *OperatingDayCount* MINUS the number of calendar days in the quarter being checked that are PRIOR to the ***PriorRATAEventRecord***.QACertEventDate

If (*OperatingDayCount* is less than the number of calendar days in the quarter being checked that are ON OR AFTER the ***PriorRATAEventRecord***.QACertEventDate)

Set ***PriorRATAEventRecord***.MaxOpDaysPriorQuarter = *OperatingDayCount*.

else

Set ***PriorRATAEventRecord***.MaxOpDaysPriorQuarter = the number of calendar days in the quarter being checked that are ON OR AFTER the ***PriorRATAEventRecord***.QACertEventDate.

else

Set ***PriorRATAEventRecord***.MinOpDaysPriorQuarter = ***PriorRATAEventRecord***.MinOpDaysPriorQuarter + *OperatingDayCount*.
Set ***PriorRATAEventRecord***.MaxOpDays PriorQuarter = ***PriorRATAEventRecord***.MaxOpDaysPriorQuarter + *OperatingDayCount*.

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusSystemTypeCode*** is “NOX”)

Set *CurrentOpDays* to Days in ***SystemOperatingSuppDataDictionaryArray*** for the current location where SystemId is equal to ***QaStatusSystemId***.

Else

Set *CurrentOpDays* to ***Rpt Period Op Days Accumulator Array*** for the Location.

if (***PriorRATAEventRecord***.MinOpDaysPriorQuarter == -1
set ***CurrentRATAStatus*** to "Missing Op Data"

else if (***PriorRATAEventRecord***.MinOpDaysPriorQuarter + *CurrentOpDays* > 90)

Set ***CurrentRATAStatus*** = "OOC-Conditional Period Expired".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = 1.0.

else if (***PriorRATAEventRecord***.MinOpDaysPriorQuarter == ***PriorRATAEventRecord***.MaxOpDaysPriorQuarter)
Set ***CurrentRATAStatus*** = "IC-Conditional".

else if (***PriorRATAEventRecord***.MaxOpDaysPriorQuarter + *CurrentOpDays* > 90)
Set ***CurrentRATAStatus*** = "Undetermined-Conditional Data".

else
Set ***CurrentRATAStatus*** = "IC-Conditional".

else
Set ***CurrentRATAStatus*** = "IC-Conditional".

else

If (the quarter of the ***PriorRATAEventRecord***.ConditionalBeginDate is equal to the quarter of the ***CurrentDateHour***)

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusSystemTypeCode*** is “NOX”)

If (Hours in ***QaCertEventSuppDataDictionaryArray*** for the current location and Conditional Data Begin Hour where QaCertEventKey is equal to ***PriorRATAEventRecord***.QaCertEventKey > 720)
Set ***CurrentRATAStatus*** = "OOC-Conditional Period Expired".
Else
Set ***CurrentRATAStatus*** = "IC-Conditional".
Else

Count the number of ***HourlyOpData*** records for the location where OpTime is greater than 0 and Date/Hour is ON OR AFTER the ***PriorRATAEventRecord***.ConditionalBeginDate/Hour and ON OR BEFORE ***CurrentDateHour***,

If the number > 720,

Set ***CurrentRATAStatus*** = "OOC-Conditional Period Expired".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = 1.0.

else

Set ***CurrentRATAStatus*** = "IC-Conditional".

else

if (***PriorRATAEventRecord***.MinOpHoursPriorQuarter is null)
Set ***PriorRATAEventRecord***.MinOpHoursPriorQuarter = 0
Set ***PriorRATAEventRecord***.MaxOpHoursPriorQuarter = 0

for each quarter beginning with the quarter of the ***PriorRATAEventRecord***.ConditionalBeginDate and continuing through the quarter BEFORE the ***CurrentDateHour***:

if (***EarliestLocationReportDate*** <= the last day of the quarter being checked)

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusSytemTypeCode*** is “NOX”)

Locate a record in ***SystemOperatingSuppDataRecordsByLocation*** where:

1) SystemId is equal to ***QaStatusSystemId***.
2) Year is equal to the year being checked.
3) Quarter is equal to the quarter being checked.
4) OpSuppDataTypeCode = "OP" if ***AnnualReportingRequirement*** == true OR the quarter being checked != 2, otherwise “OPMJ”.

If (***SystemOperatingSuppDataRecordsByLocation*** is found)
Set OperatingHourCount = ***SystemOperatingSuppDataRecordsByLocation***.Hours.
Else
Set OperatingHourCount = null.

Else

if (***Annual Reporting Requirement*** == false AND the quarter being checked == 2)

Locate the record in ***OperatingSuppDataRecordsbyLocation*** where the OpTypeCode is equal to "OSHOURS" and the reporting period is equal to the quarter being checked.

else

Locate the record in ***OperatingSuppDataRecordsbyLocation*** where the OpTypeCode is equal to "OPHOURS", FuelCode is null, and the reporting period is equal to the quarter being checked.

If (***OperatingSuppDataRecordsByLocation*** is found)
Set *OperatingHourCount* = ***OperatingSuppDataRecordsByLocation***.OpValue.
Else
Set *OperatingHourCount* = null.

if (*OperatingHourCount* is null)

Locate the record in ***ReportingFrequencyByLocation*** where CalendarYear/Quarter are equal to the quarter being checked.

if found, and (the quarter being checked is 2 or 3, or ***ReportingFrequencyByLocation***.ReportingFrequencyCode is equal to "Q"),

Set ***PriorRATAEventRecord***.MinOpHoursPriorQuarter = -1
Set ***RATAMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
exit for.

else if (the quarter being checked is the quarter of the ***PriorRATAEventRecord***.ConditionalBeginDate)

supplementalCount = null.

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusSystemTypeCode*** is “NOX” AND ***PriorRATAEventRecord***.ConditionalBeginHourSystemSuppDataExists is true)

supplementalCount = ***PriorRATAEventRecord***.ConditionalBeginSystemOpHoursCount.

If (*supplementalCount* is null AND ***PriorRATAEventRecord***.ConditionalBeginHourSuppDataExists is true)

supplementalCount = ***PriorRATAEventRecord***.ConditionalBeginOpHoursCount.

If (*supplementalCount* is NOT null)

Set ***PriorRATAEventRecord***.MinOpHoursPriorQuarter = ***PriorRATAEventRecord***.MinOpHoursPriorQuarter + *supplementalCount*.
Set ***PriorRATAEventRecord***.MaxOpHoursPriorQuarter = ***PriorRATAEventRecord***.MaxOpHoursPriorQuarter + *supplementalCount*.

Else

If (*OperatingHourCount* MINUS the number of calendar hours in the quarter being checked that are PRIOR to the ***PriorRATAEventRecord***.ConditionalBeginDate/Hour > 0)

Set ***PriorRATAEventRecord***.MinOpHoursPriorQuarter = *OperatingHourCount* MINUS the number of calendar hours in the quarter being checked that are PRIOR to the ***PriorRATAEventRecord***.ConditionalBeginDate/Hour

If (*OperatingHourCount* is less than the number of calendar hours in the quarter begin checked that are ON OR AFTER the ***PriorRATAEventRecord***.ConditionalBeginDate/Hour)

Set ***PriorRATAEventRecord***.MaxOpHoursPriorQuarter = *OperatingHourCount*.

else

Set ***PriorRATAEventRecord***.MaxOpHoursPriorQuarter = the number of calendar hours in the quarter being checked that are ON OR AFTER the ***PriorRATAEventRecord***.ConditionalBeginDate/Hour.

else

Set ***PriorRATAEventRecord***.MinOpHoursPriorQuarter = ***PriorRATAEventRecord***.MinOpHoursPriorQuarter + *OperatingHourCount*.
Set ***PriorRATAEventRecord***.MaxOpHours PriorQuarter = ***PriorRATAEventRecord***.MaxOpHoursPriorQuarter + *OperatingHourCount*.

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusSystemTypeCode*** is “NOX”)

Set *CurrentOpHours* to Hours in ***SystemOperatingSuppDataDictionaryArray*** for the current location where SystemId is equal to ***QaStatusSystemId***.

Else

Set *CurrentOpHours* to ***Rpt Period Op Hours Accumulator Array*** for the Location.

if (***PriorRATAEventRecord***.MinOpHoursPriorQuarter == -1)

set ***CurrentRATAStatus*** to "Missing Op Data"

else if (***Rpt Period Op Days Accumulator Array*** for the location == -1)

if (***PriorRATAEventRecord***.MinOpHoursPriorQuarter > 720)

Set ***CurrentRATAStatus*** = "OOC-Conditional Period Expired".

if (***CurrentMhvParameter*** == "FLOW")

Set ***OverrideRATABAF*** = 1.0.

else

Set ***CurrentRATAStatus*** = "Invalid Op Data".

else
if (***PriorRATAEventRecord***.MinOpHoursPriorQuarter + *CurrentOpHours* > 720)

Set ***CurrentRATAStatus*** = "OOC-Conditional Period Expired".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = 1.0.

else if (***PriorRATAEventRecord***.MinOpHoursPriorQuarter == ***PriorRATAEventRecord***.MaxOpHoursPriorQuarter)
Set ***CurrentRATAStatus*** = "IC-Conditional".

else if (***PriorEventRecord***.MaxOpHoursPriorQuarter + *CurrentOpHours* > 720)
Set ***CurrentRATAStatus*** = "Undetermined-Conditional Data".

else
Set ***CurrentRATAStatus*** = "IC-Conditional".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2/O2 RATA Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- H2O RATA Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- H2OM RATA Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- Hg RATA Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOX RATA Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- NOXC RATA Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB NOX RATA Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 RATA Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report --------- Stack Flow RATA Status Evaluation

Check Code: RATSTAT-6

Check Name: Evaluate Prior Multi-Level RATA

Related Former Checks:

Applicability: CEM Check

Description: Determines (if applicable) if a prior multi level RATA exists and is acceptable.

Specifications:

Set ***PriorRataIsAlternateSingleLevelRATA*** = false.
Set ***ThreeLoadRATAExpirationDate*** to null.

If (***CurrentMhvParameter*** == "FLOW" AND ***PriorRATARecord*** is not null AND ***CurrentRATAStatus*** is null, starts with "IC", or starts with "Undetermined")

Set ***PriorRataIsAlternateSingleLevelRATA*** = true.
Set ***PriorMultiLevelRATARecord*** = null.
Set ***InvalidMultiLevelRATARecord*** = null.
Set ***PriorMaxLevelRATARecord*** = null.

if (the number of levels in ***PriorRATARecord***.OpLevelCodeList is greater than or equal to the ***MaxLevelCount***)

Set ***PriorRataIsAlternateSingleLevelRATA*** = false.
exit check.

else if (***PriorRATARecord***.OpLevelCodeList contains 2 levels)

Set ***PriorRataIsAlternateSingleLevelRATA*** = false.

if (***MaxLevelCount*** <> 3)

exit check.

else if (***AnnualReportingRequiremnt*** == true)

if (***PriorRATARecord***.OpLevelCodeList contains 1 level and ***PriorRATARecord***.TestClaimCode == "SLC")

Set ***PriorRataIsAlternateSingleLevelRATA*** = false.

if (***PriorRataIsAlternateSingleLevelRATA*** == true AND ***RATAEventOperatingLevelCount*** is null or less than 2)

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***PriorRATARecord***.SystemID and the TestResultCode is not equal to "INVALID" and the EndDate/Hour is prior to the ***PriorRATARecord***.EndDate/Hour, and (the number of operating levels in the OpLevelCodeList is greater than or equal to 2 or the TestClaimCode == "SLC").

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***PriorMultiLevelRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***PriorRATARecord***.SystemID and the EndDate/Hour is prior to the ***PriorRATARecord***.EndDate/Hour and the EndDate/Hour is greater than the ***PriorMultiLevelRATARecord***.EndDate/Hour and the TestResult is equal to "INVALID", and (the number of operating levels the OpLevelCodeList is greater than or equal to 2 or the TestClaimCode == "SLC").

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***InvalidMultiLevelRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

else

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***PriorRATARecord***.SystemID and the EndDate/Hour is prior to the ***PriorRATARecord***.EndDate/Hour and the TestResultCode is equal to "INVALID", and the number of operating levels the OpLevelCodeList is greater than or equal to 2 or the TestClaimCode == "SLC").

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***InvalidMultiLevelRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

if (***PriorMultiLevelRATARecord*** is not null)

Locate the most recent record in ***QACertificationEventRecords*** where the SystemID is equal to the ***PriorMultiLevelRATARecord***.SystemID and RATA2Required is equal to "Y" and the QACertEventDate is either:

a) prior to the ***CurrentDateHour*** OR
b) equal to both the ***CurrentDateHour*** and the ConditionalBeginDate/Hour;

AND either:

a) QACertEventDate/Hour is after the ***PriorMultiLevelRATARecord***.EndDate/Hour OR
b) QACertEventDate/Hour is equal to the ***PriorMultiLevelRATARecord***.EndDate/Hour AND (TestCompletionDate is null or the TestCompletionDate/Hour is after the ***PriorMultiLevelRATARecord***.EndDate/Hour)

AND either

a) ***Annual Reporting Requirement*** is equal to true OR
b) QACertEventDate/Hour is on or after April 1 of the year of ***CurrentDateHour***

if (***QACertificationEventRecords*** is found)

Set ***SubsequentRATARecord*** = ***PriorMultiLevelRATARecord***.
Set ***CurrentRataStatus*** = "OOC-Incomplete Recertification".
Set ***OverrideRataBaf*** = 1.0.

else

if (***PriorMultiLevelRATARecord*** .QANeedsEvaluationFlag = "Y")

Set ***CurrentRATAStatus*** = "Prior Multi-Level RATA Not Yet Evaluated".

else if (***PriorMultiLevelRATARecord*** .TestResultCode = null or ***PriorMultiLevelRATARecord*** .TestResultCode = "FAILED" or ***PriorMultiLevelRATARecord*** .TestResultCode = "ABORTED")

Set ***CurrentRataStatus*** = "OOC-Incomplete QA RATA".
Set ***OverrideRataBaf*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

else if (the number of levels in ***PriorMultiLevelRATARecord***.OpLevelCodeList is greater than or equal to ***MaxLevelCount***)

exit check.

else

Set ***CurrentRataStatus*** = "OOC-Incomplete QA RATA".
Set ***OverrideRataBaf*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

if (***CurrentRATAStatus*** is null, starts with "IC", or starts with "Undetermined" AND ***RATAEventOperatingLevelCount*** is null or less than ***MaxLevelCount***))

Set ***InvalidMultiLevelRATARecord*** = null.

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***QaStatusSystemId*** and the TestResultCode is not equal to "INVALID" and the EndDate/Hour is prior to the ***PriorRATARecord***.EndDate/Hour and the number of operating levels in the OpLevelCodeList is equal to the ***MaxLevelCount***.

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***PriorMaxLevelRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***QaStatusSystemId*** and the EndDate/Hour is prior to the ***PriorRATARecord***.EndDate/Hour and the EndDate/Hour is greater than the ***PriorMaxLevelRATARecord***.EndDate/Hour and the TestResultCode is equal to "INVALID" and the number of operating levels the OpLevelCodeList is equal to the ***MaxLevelCount***.

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***InvalidMultiLevelRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

else

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***QaStatusSystemId*** and the EndDate/Hour is prior to the ***PriorRATARecord***.EndDate/Hour and the TestResultCode is equal to "INVALID" and the number of operating levels the OpLevelCodeList is equal to the ***MaxLevelCount***.

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***InvalidMultiLevelRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus***.

if (***PriorMaxLevelRATARecord*** is null)

Set ***CurrentRATAStatus*** = "OOC-No Prior Maximum Level RATA".
Set ***OverrideRataBaf*** = 1.0.

else

Locate the most recent record in ***QACertificationEventRecords*** where the SystemID is equal to the ***QaStatusSystemId*** and

AND either

a) MaxLevelCount is equal to 2 and RATA2Required is equal to "Y" OR
b) MaxLevelCount is equal to 2 and RATA3Required is equal to "Y" OR
c) MaxLevelCount is equal to 3 and RATA3Required is equal to "Y"

AND the QACertEventDate is either:

a) prior to the ***CurrentDateHour***OR
b) equal to both the ***CurrentDateHour***and the ConditionalBeginDate/Hour;

AND either:

a) QACertEventDate/Hour is after the ***PriorMaxLevelRATARecord***.EndDate/Hour OR
b) QACertEventDate/Hour is equal to the ***PriorMaxLevelRATARecord***.EndDate/Hour AND (TestCompletionDate is null or the TestCompletionDate/Hour is after the ***PriorMaxLevelRATARecord***.EndDate/Hour)

if (***QACertificationEventRecords*** is found)

Set ***SubsequentRATARecord*** = ***PriorMaxLevelRATARecord***.
Set ***CurrentRataStatus*** = "OOC-Incomplete Recertification".
Set ***OverrideRataBaf*** = 1.0.

else if (***PriorMaxLevelRATARecord***.QANeedsEvaluationFlag = "Y")

Set ***CurrentRATAStatus*** = "Prior Maximum Level RATA Not Yet Evaluated".

else if (***PriorMaxLevelRATARecord***.TestResultCode = null)

Set ***CurrentRATAStatus*** = "OOC-Prior Maximum Level RATA Has Critical Errors".
Set ***OverrideRataBaf*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

else if (***PriorMaxLevelRATARecord***.TestResultCode = "FAILED")

Set ***CurrentRATAStatus*** = "OOC-Prior Maximum Level RATA Failed".
Set ***OverrideRataBaf*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

else if (***PriorMaxLevelRATARecord***.TestResultCode = "ABORTED")

Set ***CurrentRATAStatus*** = "OOC-Prior Maximum Level RATA Aborted".
Set ***OverrideRataBaf*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

else

if ***PriorMaxLevelRATARecord***.TestReasonCode = "INITIAL",

Locate the latest record in ***QACertificationEventRecords*** where

1) SystemID is equal to the ***QaStatusSystemId***
2) QaCertEventCode is equal to "305"
3) QACertEventDate/Hour is prior to the ***CurrentDateHour***;

if (***QACertificationEventRecords*** is found, and the TestCompletionDate in the located record is after ***PriorMaxLevelRATARecord***.EndDate)
Set ***PriorMaxLevelRATARecord***.TestExpirationDate = the end of the quarter twenty quarters after the TestCompletionDate .
else
Set ***PriorMaxLevelRATARecord***.TestExpirationDate = the end of the quarter twenty quarters after the ***PriorMaxLevelRATARecord***.EndDate.
else
Set ***PriorMaxLevelRATARecord***.TestExpirationDate = the end of the quarter twenty quarters after the ***PriorMaxLevelRATARecord***.EndDate.

if (the date for ***CurrentDateHour*** is after the ***PriorMaxLevelRATARecord***.TestExpirationDate)

if (***Annual Reporting Requirement*** == false)

Set ***CurrentRATAStatus*** = "OOC-Prior Maximum Level RATA Expired".
Set ***OverrideRataBaf*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

else

Set *GraceOpHours* = ***RptPeriodOpHoursAccumulatorArray*** for the location.

If (*GraceOpHours* < 0)

Set ***CurrentRATAStatus*** = "Invalid Op Data".

else if (*GraceOpHours* > 720)

Set ***CurrentRATAStatus*** = "OOC-Prior Maximum Level RATA Expired".
Set ***OverrideRataBaf*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

else

For each quarter beginning with the quarter after the ***PriorMaxLevelRATARecord***.TestExpirationDate and continuing through the quarter prior to the ***CurrentDateHour***,

if (***EarliestLocationReportDate*** <= the last day of the quarter being checked)

Locate a record in ***OperatingSuppDataRecordsByLocation*** where the reporting period is equal to the year/quarter being checked and the OpTypeCode = "OPHOURS" and FuelCode is null.

if (***OperatingSuppDataRecordsByLocation*** is found)

Add OpValue to *GraceOpHours.*

if (*GraceOpHours* > 720)

Set ***CurrentRATAStatus*** = "OOC-Prior Maximum Level RATA Expired".
Set ***OverrideRataBaf*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

exit for.

else

Locate the record in ***ReportingFrequencyByLocation*** where CalendarYear/Quarter are equal to the quarter being checked.

if found, and (the quarter being checked is 2 or 3, or ***ReportingFrequencyByLocation***.ReportingFrequencyCode is equal to "Q"),

Set ***CurrentRATAStatus*** = "Missing Op Data".

Set ***RATAMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
exit for.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2/O2 RATA Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- H2O RATA Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- H2OM RATA Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- Hg RATA Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOX RATA Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- NOXC RATA Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB NOX RATA Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 RATA Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report --------- Stack Flow RATA Status Evaluation

Check Code: RATSTAT-7

Check Name: Determine Expiration Dates for Most Recent Prior RATA Test

Related Former Checks:

Applicability: CEM Check

Description: Determines the expiration dates for the Applicable Prior RATA test. This includes the Test Expiration Date both with and without any extensions

Specifications:

if (***CurrentRATAStatus*** is null and ***PriorRATARecord*** is not null and ***PriorRATAEventRecord*** is null)

Set *PriorTestExpirationDate* = null
Set *PriorTestExpirationDateWithExtension* = null
Set *MissingOpData* = false
Set *NumberOfExtensionQuarters* = 0

Set *PriorTestExpirationDate* = ***PriorRATARecord***.TestExpirationDate.
Set *PriorTestExpirationDateWithExtension* = ***PriorRATARecord***.TestExpirationDateWithExtension.

If (***PriorRATARecord***.IgnoreGraceForExtensions is equal to 1)
Set *PriorTestIgnoreGraceForExtensions* = true
Else
Set *PriorTestIgnoreGraceForExtensions* = false

if (*PriorTestExpirationDate* is null)

if (***Annual Reporting Requirement*** == false)

if (***PriorRATARecord***.EndDate is between 10/01/2007 and 12/31/2007)
Set *PriorTestExpirationDate* = 09/30/2008
else
Set *PriorTestExpirationDate* = September 30th of the year of the ***PriorRATARecord***.EndDate.

else if (***QaStatusSystemDesignationCode*** == "B")

Locate a record in ***TestExtensionExemptionRecords*** where the SystemID is equal to the ***QaStatusSystemId*** and the ExtensionExemptionCode is equal to "NRB720" and a ReportingPeriod equal to the ***CurrentReportingPeriod***.

if (***TestExtensionExemptionRecords*** is found)

Set *PriorTestExpirationDate* = the end of the quarter eight quarters after the ***PriorRATARecord***.EndDate.

if (*PriorTestExpirationDate* is null)

if (***PriorRataIsAlternateSingleLevelRATA*** == true)

Set *PriorTestExpirationDate* = the end of the quarter one year after the ***PriorMultiLevelRATARecord***.EndDate.

if (***PriorMultiLevelRATARecord***.GracePeriodInd == 1)

Set *PriorTestExpirationDate* = the end of the quarter prior to the *PriorTestExpirationDate*.

else
if (***PriorRATARecord***.RataFrequencyCd in set {4QTRS,8QTRS})

Locate the most recent ***QACertificationEventRecords*** for the ***PriorRATARecord***.SystemID where RATARequired is equal to "Y" and the BeginDate/Hour is prior to the ***PriorRATARecord***.BeginDate/Hour.

if (***QACertificationEventRecords*** is found and RATACertEvent == "Y" and the ConditionalDataBeginDate is null and the CompletionTestDate/Hour is after the ***PriorRATARecord***.EndDate/Hour)

if (***PriorRATARecord***.SystemTypeCode is in set (HCL, HF, HG, ST))

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the ProgramCode is equal to MATS and the EmissionsRecordingBeginDate is ON OR BEFORE the BeginDate of the associated system in the ***PriorRATAEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is found) and (EmissionsRecordingBeginDate is later than ***QACertificationEventRecords***.CompletionTestDate)

Set *PriorTestExpirationDate* = the end of the quarter one year after the EmissionsRecordingBeginDate.

Else

Set *PriorTestExpirationDate* = the end of the quarter one year after the ***QACertificationEventRecords***.CompletionTestDate.

if (***PriorRATARecord***.GracePeriodInd == 1)

Set *PriorTestIgnoreGraceForExtensions* = true.

Else

Set *PriorTestExpirationDate* = the end of the quarter one year after the ***QACertificationEventRecords***.CompletionTestDate.

if (***PriorRATARecord***.GracePeriodInd == 1)

Set *PriorTestExpirationDate* = the end of the quarter prior to the *PriorTestExpirationDate.*

else

if (***PriorRATARecord***.SystemTypeCode is in set (HCL, HF, HG, ST))

Locate the record in ***LocationProgramRecordsByHourLocation*** with the latest EmissionsRecordingBeginDate where the ProgramCode is equal to MATS and the EmissionsRecordingBeginDate is ON OR BEFORE the BeginDate of the associated system in the ***PriorRATAEventRecord***.

If (the record in ***LocationProgramRecordsByHourLocation*** is found) and (EmissionsRecordingBeginDate is later than ***PriorRATARecord***.EndDate)

Set *PriorTestExpirationDate* = the end of the quarter one year after the EmissionsRecordingBeginDate.

Else

Set *PriorTestExpirationDate* = the end of the quarter one year after the ***PriorRATARecord***.EndDate.

Else

Set *PriorTestExpirationDate* = the end of the quarter one year after the ***PriorRATARecord***.EndDate.

if (***PriorRATARecord***.GracePeriodInd == 1)

Set *PriorTestExpirationDate* = the end of the quarter prior to the *PriorTestExpirationDate.*

else

Locate the most recent ***QACertificationEventRecords*** for the ***PriorRATARecord***.SystemID where RATARequired is equal to "Y" and the BeginDate/Hour is prior to the ***PriorRATARecord***.BeginDate/Hour.

if (***QACertificationEventRecords*** is found AND RATACertEvent == "Y" and the ConditionalDataBeginDate is null and the CompletionTestDate/Hour is after the ***PriorRATARecord***.EndDate/Hour)

Set *PriorTestExpirationDate* = the end of the quarter two quarters after the ***QACertificationEventRecords***.CompletionTestDate.

else

Set *PriorTestExpirationDate* = the end of the quarter two quarters after the ***PriorRATARecord***.EndDate.

Set ***PriorRATARecord***.TestExpirationDate = *PriorTestExpirationDate.*

If (*PriorTestIgnoreGraceForExtensions* is true)
Set ***PriorRATARecord***.IgnoreGraceForExtensions = 1
Else
Set ***PriorRATARecord***.IgnoreGraceForExtensions = 0

if (***CurrentDateHour*** is ON OR BEFORE the *PriorTestExpirationDate*)

Set ***CurrentRATAStatus*** = "IC".

else if (***Annual Reporting Requirement*** == false)

Set ***CurrentRATAStatus*** = "OOC-Expired".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

else

if (*PriorTestExpirationDateWithExtension* is null)

if (***CurrentMhvParameter*** == "FLOW" and ***PriorRataIsAlternateSingleLevelRATA*** == true)

if (***PriorMultiLevelRATARecord***.GracePeriodInd == 1)
StartQuarter = the quarter of the ***PriorMultiLevelRATARecord***.EndDate
else
StartQuarter = the quarter after the ***PriorMultiLevelRATARecord***.EndDate

set *EndQuarter* = the quarter two years after the quarter of the ***PriorMultiLevelRATARecord***.EndDate.

else

if (***PriorRATARecord***.GracePeriodInd == 1) and (*PriorTestIgnoreGraceForExtensions* is false)
StartQuarter = the quarter of the ***PriorRATARecord***.EndDate
else
StartQuarter = the quarter after the ***PriorRATARecord***.EndDate

set *EndQuarter* = the quarter two years after the quarter of the ***PriorRATARecord***.EndDate.

Set *MaximumExtensionDate* = the last date of *EndQuarter*
Set *StartNonQaPrimaryBypassQuarter* = *StartQuarter*

For each quarter beginning with the *StartQuarter* and continuing through the earlier of the quarter prior to the quarter of the ***CurrentDateHour*** and *EndQuarter*

// Prevent extensions beyond the maximum expiration date
if *PriorTestExpirationDate* plus *NumberOfExtensionQuarters* + 1 is after *MaximumExtensionDate*,
exit loop

if (***EarliestLocationReportDate*** > the last day of the quarter being checked)

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.
Set *StartNonQaPrimaryBypassQuarter* = year/quarter being checked plus one quarter.

else

If (PrimaryBypassActiveForHour is true AND QaStatusSystemTypeCode is “NOX”)

Locate a record in SystemOperatingSuppDataRecordsByLocation where:

1) SystemId is equal to QaStatusSystemId.
2) Year is equal to the year being checked.
3) Quarter is equal to the quarter being checked.
4) OpSuppDataTypeCode = "OP".

If (SystemOperatingSuppDataRecordsByLocation is found)
Set OperatingHourCount =
SystemOperatingSuppDataRecordsByLocation.Hours.
Else
Set OperatingHourCount = null.

Else

Locate a record in ***OperatingSuppDataRecordsByLocation*** where the reporting period is equal to the year/quarter being checked and the OpTypeCode = "OPHOURS".

If (***OperatingSuppDataRecordsbyLocation*** is found)
Set *OperatingHourCount* = ***OperatingSuppDataRecordsByLocation***.OpValue.
Else
Set *OperatingHourCount* = null.

if (*OperatingHourCount* is NOT null AND *OperatingHourCount* < 168)

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.
Set *StartNonQaPrimaryBypassQuarter* = year/quarter being checked plus one quarter.

else if (***QaStatusSystemTypeCode*** begins with "SO2")

Locate a record in ***TestExtensionExemptionRecords*** where the SystemID is equal to the ***QaStatusSystemId*** and the ExtensionExemptionCode is equal to "LOWSQTR" and the reporting period is equal to the year/quarter being checked.

if (***TestExtensionExemptionRecords*** is found)

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.
Set *StartNonQaPrimaryBypassQuarter* = year/quarter being checked plus one quarter.

else if (*OperatingHourCount* is null)

Locate the record in ***ReportingFrequencyByLocation*** where CalendarYear/Quarter are equal to the year/quarter being checked .

if found, and (the quarter being checked is 2 or 3, or ***ReportingFrequencyByLocation***.ReportingFrequencyCode is equal to "Q"),

Set *Missing Op Data* to true.
Set ***RATAMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)

else if (***QaStatusSystemDesignationCode*** == "PB")

Locate a record in ***TestExtensionExemptionRecords*** where the SystemID is equal to the ***QaStatusSystemId***, the ExtensionExemptionCode is equal to "NONQAPB" or "GRACEPB", and the reporting period is equal to the year/quarter being checked.

if (***TestExtensionExemptionRecords*** is found)

Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.
Set *StartNonQaPrimaryBypassQuarter* = year/quarter being checked plus one quarter.

else if (*OperatingHourCount* is null)

Locate the record in ***ReportingFrequencyByLocation*** where CalendarYear/Quarter are equal to the year/quarter being checked .

if found, and (the quarter being checked is 2 or 3, or ***ReportingFrequencyByLocation***.ReportingFrequencyCode is equal to "Q"),

Set *Missing Op Data* to true.
Set ***RATAMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)

else if (*OperatingHourCount* is null)

Locate the record in ***ReportingFrequencyByLocation*** where CalendarYear/Quarter are equal to the year/quarter being checked .

if found, and (the quarter being checked is 2 or 3, or ***ReportingFrequencyByLocation***.ReportingFrequencyCode is equal to "Q"),

Set *Missing Op Data* to true.
Set ***RATAMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)

if (***QaStatusSystemDesignationCode*** == "PB")

// Allow additional extensions for non QA Primary Bypass exemptions.
For each quarter beginning with *StartNonQaPrimaryBypassQuarter* and continuing through the quarter prior to the quarter of the ***CurrentDateHour***

Locate a record in ***TestExtensionExemptionRecords*** where the SystemID is equal to the ***QaStatusSystemId***, the ExtensionExemptionCode is equal to "NONQAPB" or "GRACEPB", and the reporting period is equal to the year/quarter being checked.

if (***TestExtensionExemptionRecords*** is found)
Set *NumberOfExtensionQuarters* = *NumberOfExtensionQuarters* + 1.
else
exit loop.

Set *PriorTestExpirationDateWithExtension* = *PriorTestExpirationDate*.
Add *NumberOfExtensionQuarters* to *PriorTestExpirationDateWithExtension*.

If *PriorTestExpirationDateWithExtension* is greater than *MaximumExtensionDate*
Set PriorTestExpirationDateWithExtension = MaximumExtensionDate

Set ***PriorRATARecord***.TestExpirationDateWithExtension = *PriorTestExpirationDateWithExtension*.

else

Set ***PriorRATARecord***.TestExpirationDateWithExtension = *PriorTestExpirationDate*

If (***CurrentDateHour*** is ON OR BEFORE the *PriorTestExpirationDateWithExtension*)

Set ***CurrentRATAStatus*** = "IC-Extension".

else if (*Missing Op Data* is true)

Set ***CurrentRATAStatus*** = "Missing Op Data".
Set ***PriorRATARecord***.TestExpirationDateWithExtension = null

else

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusSystemTypeCode*** is “NOX”)

Set *CurrentOpHours* = Hours in ***SystemOperatingSuppDataDictionaryArray*** for the current location where SystemId is equal to ***QaStatusSystemId***.
Else
Set *CurrentOpHours* = RptPeriodOpHoursAccumulatorArray for the location.

if (CurrentOpHours == -1)

Set ***CurrentRATAStatus*** = "Invalid Op Data".

else

Set *GraceOpHours* = *CurrentOpHours*.

if (*GraceOpHours* > 720)

Set ***CurrentRATAStatus*** = "OOC-Expired".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

else

If there are no quarters beginning with the LATER of the quarter after the *PriorTestExpirationDateWithExtension* and the quarter of the ***EarliestLocationReportDate*** and ending with the quarter prior to the ***CurrentDateHour***,

Set ***CurrentRATAStatus*** = "IC-Grace".

else

For each quarter beginning with the quarter after the *PriorTestExpirationDateWithExtension* and continuing through the quarter prior to the ***CurrentDateHour***,

if (***EarliestLocationReportDate*** <= the last day of the quarter being checked)

If (***PrimaryBypassActiveForHour*** is true AND ***QaStatusSystemTypeCode*** is “NOX”)

Locate a record in ***SystemOperatingSuppDataRecordsByLocation*** where:

1) SystemId is equal to ***QaStatusSystemId***.
2) Year is equal to the year being checked.
3) Quarter is equal to the quarter being checked.
4) OpSuppDataTypeCode = "OP".

If (***SystemOperatingSuppDataRecordsByLocation*** is found)
Set *OperatingHourCount* = ***SystemOperatingSuppDataRecordsByLocation***.Hours.
Else
Set *OperatingHourCount* = null.

Else

Locate a record in ***OperatingSuppDataRecordsByLocation*** where the reporting period is equal to the year/quarter being checked and the OpTypeCode = "OPHOURS" and FuelCode is null.

If (***OperatingSuppDataRecordsByLocation*** is found)
Set *OperatingHourCount* = ***OperatingSuppDataRecordsByLocation***.OpValue.
Else
Set *OperatingHourCount* = null.

if (*OperatingHourCount* is NOT null)

Add *OperatingHourCount* to *GraceOpHours.*

if (*GraceOpHours* > 720)

Set ***CurrentRATAStatus*** = "OOC-Expired".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

exit for.

else

Locate the record in ***ReportingFrequencyByLocation*** where CalendarYear/Quarter are equal to the year/quarter being checked .

if found, and (the quarter being checked is 2 or 3, or ***ReportingFrequencyByLocation***.ReportingFrequencyCode is equal to "Q"),

Set ***CurrentRATAStatus*** = "Missing Op Data".
Set ***RATAMissingOpDataInfo*** = "[YEAR] Q[QTR]" (where [YEAR] is the year of the quarter being checked and [QTR] is the number of the quarter being checked.)
exit for.

if (***CurrentRATAStatus*** is null)

Set ***CurrentRATAStatus*** = "IC-Grace".

If (***PriorRataIsAlternateSingleLevelRATA*** == true AND ***CurrentRATAStatus*** = "OOC-Expired")
Set ***CurrentRATAStatus*** = "OOC-Incomplete QA RATA".

if (***CurrentMhvParameter*** == "FLOW")
Set ***OverrideRATABAF*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2/O2 RATA Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- H2O RATA Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- H2OM RATA Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- Hg RATA Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOX RATA Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- NOXC RATA Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB NOX RATA Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 RATA Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report --------- Stack Flow RATA Status Evaluation

Check Code: RATSTAT-8

Check Name: Determine Final RATA Status

Related Former Checks:

Applicability: CEM Check

Description: Evaluates the determined RATA Status and changes it if needed based on an ignored test or the status of the alternate system.

Specifications:

Set ***AlternateRATARecord*** = null.

if (***CurrentRATAStatus*** begins with "OOC")

Set ***InvalidRATATestNumber*** = null.

if (***InvalidMultiLevelRATARecord*** is not null)

Set ***Invalid RATA Test Number*** = ***InvalidMultiLevelRATARecord***.TestNumber
Set ***CurrentRATAStatus*** = ***CurrentRATAStatus*** & "*".

if (***CurrentMhvParameter*** == "FLOW")
Set ***RATA Status BAF*** = ***InvalidMultiLevelRATARecord***.OverallBiasAdjustmentFactor.

else if (***InvalidRATARecord*** is not null)

Set ***InvalidRATATestNumber*** = ***InvalidRATARecord***.TestNumber
Set ***CurrentRATAStatus*** = ***CurrentRATAStatus*** & "*".

if (***CurrentMhvParameter*** == "FLOW")
Set ***RATA Status BAF*** = ***InvalidRATARecord***.OverallBiasAdjustmentFactor.

else if (***OverrideRATABAF*** is not null)

if (***CurrentMhvParameter*** == "FLOW")
Set ***RATA Status BAF*** = ***OverrideRATABAF***

else if (***CurrentRATAStatus*** begins with "IC" or "Undetermined"

If (***QaStatusSystemTypeCode***== "NOX")

Set *ComponentIDList* = null.
Set *AlternateSystemIDList* = null.

For each record in ***MonitorSystemComponentRecordsforHourandLocation*** where the SystemID is equal to the ***QaStatusSystemId*** and ComponentTypeCd in list {"CO2", "NOX", "O2"}

Add ***MonitorSystemComponentRecordsforHourandLocation***.ComponentID to *ComponentIDList*.

if (*ComponentIDList* is not null)

For each record in ***MonitorSystemComponentRecordsforHourandLocation*** where the ComponentID is in *ComponentIDList* and SysTypeCd in list {"CO2", "O2", "NOXC"}.

Add ***MonitorSystemComponentRecordsforHourandLocation***.SystemID to *AlternateSystemIDList.*

if (*AlternateSystemIDList* is not null)

If (***PriorRATAEventRecord*** is not null)

If (***PriorRATAEventRecord***.ConditionalBeginDate is not null)

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is in *AlternateSystemIDList* and the EndDate/Hour is prior to the ***CurrentDateHour*** and the EndDate/Hour is after the ***PriorRATAEventRecord***.ConditionalBeginDate/Hour.

If there are multiple records found with the same begindate, prefer the record with TestResultCode = "PASSED" or "PASSAPS", then prefer the record with TestResultCode = "FAILED" or "ABORTED".

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***AlternateRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus.***

if (***AlternateRATARecord***.TestResultCode = null)

Set ***CurrentRATAStatus*** = "OOC-Prior Alternate System RATA Has Critical Errors".

else if (***AlternateRATARecord***.TestResultCode = "FAILED")

Set ***CurrentRATAStatus*** = "OOC-Prior Alternate System RATA Failed".

else if (***AlternateRATARecord***.TestResultCode = "ABORTED")

Set ***CurrentRATAStatus*** = "OOC-Prior Alternate System RATA Aborted".

else

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is in *AlternateSystemIDList* and the EndDate/Hour is prior to the ***CurrentDateHour*** and the EndDate/Hour is after the ***PriorRATAEventRecord***.ConditionalBeginDate/Hour and the QANeedsEvaluationFlag is equal to "Y".

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***AlternateRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus.***
Set ***CurrentRATAStatus*** = "Prior Alternate System RATA Not Yet Evaluated".

else if (***PriorRATARecord*** is not null)

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is in *AlternateSystemIDList* and the EndDate/Hour is prior to the ***CurrentDateHour*** and the EndDate/Hour is after the ***PriorRATARecord***.EndDate/Hour.

If there are multiple records found with the same begindate, prefer the record with TestResultCode = "PASSED" or "PASSAPS", then prefer the record with TestResultCode = "FAILED" or "ABORTED".

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***AlternateRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus.***

if (***AlternateRATARecord***.TestResultCode = null)

Set ***CurrentRATAStatus*** = "OOC-Prior Alternate System RATA Has Critical Errors".

else if (***AlternateRATARecord***.TestResultCode = "FAILED")

Set ***CurrentRATAStatus*** = "OOC-Prior Alternate System RATA Failed".

else if (***AlternateRATARecord***.TestResultCode = "ABORTED")

Set ***CurrentRATAStatus*** = "OOC-Prior Alternate System RATA Aborted".

else

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is in *AlternateSystemIDList* and the EndDate/Hour is prior to the ***CurrentDateHour*** and the EndDate/Hour is after the ***PriorRATARecord***.EndDate/Hour and the QANeedsEvaluationFlag is equal to "Y".

if (***RATATestRecordsByLocationForQAStatus*** is found)

Set ***AlternateRATARecord*** = the found record in ***RATATestRecordsByLocationForQAStatus.***
Set ***CurrentRATAStatus*** = "Prior Alternate System RATA Not Yet Evaluated".

If (***Current RATA Status*** begins with "IC" or "Undetermined")

If (***PriorRATARecord*** is null)

Set ***RATA Status BAF*** = 1

else if (***Current RATA Status*** begins with "IC-Cond", "Undetermined-Cond" or "PendingOOC-Cond" AND
(***PriorRATAEventRecord***.QACertEventCode is in set {40, 50, 51, 100, 101, 120, 125, 151, 250, 255, 300, 305}) OR ***PriorRATARecord***.TestResultCode does NOT begin with "PASS"))

Set ***RATA Status BAF*** = 1

else if (***QaStatusSystemId*** is NOT equal to ***PriorRATARecord***.SystemID)

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***QaStatusSystemId***, the TestResultCode is NOT equal to "INVALID" and the EndDate/Hour is prior to the ***CurrentDateHour***

If found,
Set ***RATA Status BAF*** to the OverallBiasAdjustmentFactor in the RATA record found above.

else
Set ***CurrentRATAStatus*** to "OOC-No Prior Test or Event".

Locate the most recent record in ***RATATestRecordsByLocationForQAStatus*** for the location where the SystemID is equal to the ***QaStatusSystemId***, the TestResultCode is equal to "INVALID" and the EndDate/Hour is prior to the ***CurrentDateHour***.

If found,
Set ***InvalidRATATestNumber*** to the TestNumber in the RATA record found above.
Set ***CurrentRATAStatus*** = ***CurrentRATAStatus*** & "*".

else

Set ***RATA Status BAF*** = ***PriorRATARecord***.OverallBiasAdjustmentFactor

If (***CurrentRATAStatus*** does not begin with "IC")

Return result ***CurrentRATAStatus*** .

Results:

Result	Response	Severity
Invalid Monitor System	The [testtype] status for [key] could not be determined, because the Monitor System record for MonitoringSystemID [system] has a critical error.	Critical Error Level 1
Invalid Op Data	The [testtype] status for [key] could not be determined, because the OperatingTime in at least one Hourly Operating Data records was missing or invalid.	Critical Error Level 1
Missing Op Data	The [testtype] status for [key] could not be determined, because the Op Supp Data record for OPHOURS, OSHOURS, or OPDAYS is missing for [MISSINGOPDATAINFO] (and possibly other previous reporting periods). If you have submitted emissions data for prior quarters, you should be able to retrieve these records by logging on to the EPA host.	Critical Error Level 1
Missing Program	The [testtype] status for [key] could not be determined, because a Unit Program record associated with the initial certification event for QACertEventCode [code] QACertEventDate [eventdate] either does not exist or has a UnitMonitorCertificationBeginDate inconsistent with the BeginDate of the associated Monitor System record.	Critical Error Level 1
OOC-Conditional Period Expired	The conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for SystemID [EVENTKEY] has expired.	Critical Error Level 1
OOC-Conditional Period Expired*	The conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for SystemID [EVENTKEY] has expired.	Critical Error Level 1
OOC-Event	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for SystemID [eventkey], but you did not indicate the use of conditional data.	Critical Error Level 1
OOC-Event*	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [eventdate] for SystemID [eventkey], but you did not indicate the use of conditional data. An invalid [testtype] was ignored.	Critical Error Level 1
OOC-Expired	The prior RATA for SystemID [RATASYS] with TestNumber [testnum] has expired.	Critical Error Level 1
OOC-Expired*	The prior RATA for [RATASYS] with TestNumber [testnum] has expired. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Incomplete QA RATA	The prior RATA for FLOW SystemID [RATASYS] with TestNumber [testnum] was a single-level RATA instead of the required multi-level RATA. If applicable, please review the prior single-level RATA to determine if it contained a single load flow claim qualification record under Part 75, Appendix B, Section 2.3.1.3(c)(3)).	Critical Error Level 1
OOC-Incomplete QA RATA*	The prior RATA for FLOW SystemID [RATASYS] with TestNumber [testnum] was a single-level RATA instead of the required multi-level RATA. An invalid prior test with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Incomplete Recertification	The subsequent recertification RATA for FLOW SystemID [SUBSYS] with TestNumber [subtestnum] was performed at fewer than the required operating levels.	Critical Error Level 1
OOC-Incomplete Recertification*	The subsequent recertification RATA for FLOW SystemID [SUBSYS] with TestNumber [subtestnum] was performed at fewer than the required operating levels. An invalid prior test with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-No Prior Maximum Level RATA	You did not report a valid prior [max]-level flow RATA for [key].	Critical Error Level 1
OOC-No Prior Maximum Level RATA*	You did not report a valid prior [max]-level flow RATA for [key]. An invalid RATA with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-No Prior Test or Event	You did not report a prior [testtype] or certification event for [key].	Critical Error Level 1
OOC-No Prior Test or Event*	You did not report a valid prior [testtype] or certification event for [key]. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Prior Alternate System RATA Aborted	A prior RATA with TestNumber [alttestnum] for MonitoringSystemID [altsys] was aborted, so [key], which contains a component that is also in the former system, is out-of-control.	Critical Error Level 1
OOC-Prior Alternate System RATA Failed	A prior RATA with TestNumber [alttestnum] for MonitoringSystemID [altsys] failed, so [key], which contains a component that is also in the former system, is out-of-control.	Critical Error Level 1

OOC-Prior Alternate System RATA Has Critical Errors	A prior RATA with TestNumber [alttestnum] for MonitoringSystemID [altsys] has critical errors, so [key], which contains a component that is also in the former system, is out-of-control.	Critical Error Level 1
OOC-Prior Maximum Level RATA Aborted	The prior [max]-level flow RATA for [key] with TestNumber [maxtestnum] was aborted.	Critical Error Level 1
OOC-Prior Maximum Level RATA Aborted*	The prior [max]-level flow RATA for [key] with TestNumber [maxtestnum] was aborted. An invalid RATA with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Prior Maximum Level RATA Expired	The prior [max]-level flow RATA for [key] with TestNumber [maxtestnum] has expired.	Critical Error Level 1
OOC-Prior Maximum Level RATA Expired*	The prior [max]-level flow RATA for [key] with TestNumber [maxtestnum] has expired. An invalid RATA with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Prior Maximum Level RATA Failed	The prior [max]-level flow RATA for [key] with TestNumber [maxtestnum] failed.	Critical Error Level 1
OOC-Prior Maximum Level RATA Failed*	The prior [max]-level flow RATA for [key] with TestNumber [maxtestnum] failed. An invalid RATA with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Prior Maximum Level RATA Has Critical Errors	The prior [max]-level flow RATA for [key] with TestNumber [maxtestnum] has critical errors.	Critical Error Level 1
OOC-Prior Maximum Level RATA Has Critical Errors*	The prior [max]-level flow RATA for [key] with TestNumber [maxtestnum] has critical errors. An invalid RATA with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Prior Test Aborted	The prior RATA for SystemID [RATASYS] with TestNumber [testnum] was aborted.	Critical Error Level 1
OOC-Prior Test Aborted*	The prior RATA for SystemID [RATASYS] with TestNumber [testnum] was aborted. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Prior Test Failed	The prior RATA for SystemID [RATASYS] with TestNumber [testnum] failed.	Critical Error Level 1
OOC-Prior Test Failed*	The prior RATA for SystemID [RATASYS] with TestNumber [testnum] failed. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Prior Test Has Critical Errors	The prior RATA for SystemID [RATASYS] with TestNumber [testnum] has critical errors.	Critical Error Level 1
OOC-Prior Test Has Critical Errors*	The prior RATA for SystemID [RATASYS] with TestNumber [testnum] has critical errors. An invalid prior [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Recertification Test Aborted	The subsequent recertification RATA for SystemID [subsys] with TestNumber [subtestnum] was aborted.	Critical Error Level 1
OOC-Recertification Test Aborted*	The subsequent recertification RATA for SystemID [subsys] with TestNumber [subtestnum] was aborted. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Recertification Test Failed	The subsequent recertification RATA for SystemID [subsys] with TestNumber [subtestnum] failed.	Critical Error Level 1
OOC-Recertification Test Failed*	The subsequent recertification RATA for SystemID [subsys] with TestNumber [subtestnum] failed. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1
OOC-Recertification Test Has Critical Errors	The subsequent recertification RATA for SystemID [subsys] with TestNumber [subtestnum] has critical errors.	Critical Error Level 1
OOC-Recertification Test Has Critical Errors*	The subsequent recertification RATA for SystemID [subsys] with TestNumber [subtestnum] has critical errors. An invalid [testtype] with TestNumber [invtestnum] was ignored.	Critical Error Level 1

Prior Alternate System RATA Not Yet Evaluated	The RATA status for [key] could not be determined, because a prior RATA with TestNumber [alttestnum] for MonitoringSystemID [altsys], which contains a component that is also in the former system, has not yet been evaluated.	Critical Error Level 1
Prior Maximum Level RATA Not Yet Evaluated	The RATA status for [key] could not be determined, because the prior [max]-level flow RATA with TestNumber [maxtestnum] has not yet been evaluated.	Critical Error Level 1
Prior Multi-Level RATA Not Yet Evaluated	The RATA status could not be determined, because the prior multi-level flow RATA for SystemID [RATASYS] with TestNumber [multitestnum] has not yet been evaluated.	Critical Error Level 1
Prior Test Not Yet Evaluated	The RATA status could not be determined, because the applicable prior RATA for SystemID [RATASYS] with TestNumber [testnum] has not yet been evaluated.	Critical Error Level 1
Recertification Test Not Yet Evaluated	The RATA status could not be determined, because the subsequent recertification RATA for SystemID [subsys] with TestNumber [subtestnum] has not yet been evaluated.	Critical Error Level 1
Undetermined-Conditional Data	The software could not determine if the current hour was within the conditional data period for QACertEventCode [code] QACertEventDate [eventdate] for SystemID [eventkey].	Informational Message

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- CO2/O2 RATA Status Evaluation
2	Process/Category:	Emissions Data Evaluation Report --------- H2O RATA Status Evaluation
3	Process/Category:	Emissions Data Evaluation Report --------- H2OM RATA Status Evaluation
4	Process/Category:	Emissions Data Evaluation Report --------- Hg RATA Status Evaluation
5	Process/Category:	Emissions Data Evaluation Report --------- NOX RATA Status Evaluation
6	Process/Category:	Emissions Data Evaluation Report --------- NOXC RATA Status Evaluation
7	Process/Category:	Emissions Data Evaluation Report --------- NOXR Unused P-PB NOX RATA Status Evaluation
8	Process/Category:	Emissions Data Evaluation Report --------- SO2 RATA Status Evaluation
9	Process/Category:	Emissions Data Evaluation Report --------- Stack Flow RATA Status Evaluation

Check Category:

Weekly System Integrity Status

Check Code: WSISTAT-1

Check Name: Initialize Status Checking

Related Former Checks:

Applicability:

Description: Initialized parameters need for status checking.

Specifications:

Set ***WsiStatus*** to null.
Set ***WsiPluginEventRecord*** to null.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Hg System Integrity Status Evaluation

Check Code: WSISTAT-2

Check Name: Locate Prior Test

Related Former Checks:

Applicability:

Description: This check locates the test prior to the current hour. If a test was not found it determines whether the status is OOC because at least 7 total operating days have occurred.

Specifications:

For the ***WsiTestDictionary*** entry where the key is equal to ***QaStatusComponentId***.

If (***WsiTestDictionary*** entry exists)

Set ***WsiPriorTestRecord*** to ***WsiTestDictionary***.MostRecentTestRecord.

Else

Set ***WsiPriorTestRecord*** to null.

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Hg System Integrity Status Evaluation

Check Code: WSISTAT-3

Check Name: Check For Intervening Event

Related Former Checks:

Applicability:

Description: Locates an event with codes 110 and 130 and a matching component id, and a test date and hour preceeding the current hour but after the current test.

Specifications:

Set ***WsiInterveningEventRecord*** to null.

If (***WsiStatus*** is equal to null)

Locate the most recent record in ***QACertificationEventRecords*** where:

a) ComponentID is equal to ***QaStatusComponentId***.
b) QaCertEventDateHour is prior to ***CurrentDateHour***.
c) QaCertEventDateHour is after ***WsiPriorTestRecord***.TestDateHour.
d) QaCertEventCode is equal to "110" or "130"

If found

Set ***WsiInterveningEventRecord*** to the located record.
Set ***WsiPluginEventRecord*** to the located record.
Set ***WsiStatus*** to "OOC-Event".

Else
If (***WsiPriorTestRecord***.TestResultCode is equal to null)

Set ***WsiStatus*** to "OOC-Test Has Critical Errors".

Else if (***WsiPriorTestRecord***.TestResultCode is equal to "FAILED")

Set ***WsiStatus*** to "OOC-Test Failed".

Else

For the ***WsiTestDictionary*** entry where the key is equal to ***QaStatusComponentId***.

If (***WsiTestDictionary***.OperatingDateList is not null) AND (the count of days in ***WsiTestDictionary***.OperatingDateList is greater than 7)

Set ***WsiStatus*** to "OOC-Expired".

Else
Set ***WsiStatus*** to "IC".

Results:

Result	Response	Severity

Usage:

1	Process/Category:	Emissions Data Evaluation Report --------- Hg System Integrity Status Evaluation

Check Code: WSISTAT-4

Check Name: Return the Final Status

Related Former Checks:

Applicability:

Description: Returns the value in WSI Result Status as the check result.

Specifications:

If (***WsiStatus*** does not begin with "IC")

return result ***WsiStatus*** .

Results:

Result	Response	Severity
OOC-Event	You reported a QA Certification Event record for QACertEventCode [code] QACertEventDate [event] for [compkey], and have not yet performed the required recertification tests.	Critical Error Level 1
OOC-Expired	The prior weekly system integrity test for [compkey] on [date] has expired.	Critical Error Level 1
OOC-No Prior Test	You did not report a prior weekly system integrity test for [compkey] during the reporting period. Any weekly system integrity that may have been completed in a prior reporting period has expired.	Critical Error Level 1
OOC-Test Failed	The prior weekly system integrity test for [compkey] completed on [date] failed.	Critical Error Level 1
OOC-Test Has Critical Errors	The prior weekly system integrity test for [compkey] completed on [date] has critical errors.	Critical Error Level 1

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hg System Integrity Status Evaluation

Check Code: WSISTAT-5

Check Name: Check for Intervening Like-Kind Event

Related Former Checks:

Applicability: General Check

Description: When a previous test does not exist, this check locates an event with codes 140 and 141 and a matching component id, and an event date and hour preceeding the current hour. If found and at least seven operating days occurred between the event and the current hour, an OOC-NoPriorTest occurs. If the event is not found, and at least seven operating days have occurred since the beginning of the quarter, an OOC-NoPriorTest also occurs.

Specifications:

Set ***WsiInterveningLinkKindEventRecord*** to null.

Locate the most recent record in ***QACertificationEventRecords*** where:

a) ComponentID is equal to ***QaStatusComponentId***.
b) QaCertEventCode is equal to "140" or "141"
c) QaCertEventDateHour is prior to ***CurrentDateHour***.
d) If ***WsiPriorTestRecord*** is NOT equal to null, then QaCertEventDateHour is after ***WsiPriorTestRecord***.TestDateHour.

If found

Set ***WsiInterveningLinkKindEventRecord*** to the located record in ***QACertificationEventRecords***.

Set *EarliestOperatingDate* equal to the day after ***WsiInterveningLinkKindEventRecord***.QaCertEventDate.

If (the count of days on or after *EarliestOperatingDate* in ***OperatingDateArray*** for current location is greater than 7)
If ***WsiPriorTestRecord*** is equal to null
Set ***WsiStatus*** to "OOC-No Prior Test".
Else
Set ***WsiPluginEventRecord*** to ***WsiInterveningLinkKindEventRecord***.
Set ***WsiStatus*** to "OOC-Event".

Else
Set ***WsiStatus*** to "IC-Undetermined".

Else
If ***WsiPriorTestRecord*** is equal to null

If ***QaStatusMatsErbDate*** is not null, AND ***QaStatusMatsErbDate*** is later than ***QaStatusComponentBeginDate***,
Set *EarliestOperatingDate* equal to the day after ***QaStatusMatsErbDate*** .
Else
Set *EarliestOperatingDate* equal to the day after ***QaStatusComponentBeginDate***.

If (the count of days on or after *EarliestOperatingDate* in ***OperatingDateArray*** for current location is greater than 7)
Set ***WsiStatus*** to "OOC-No Prior Test".
Else
Set ***WsiStatus*** to "IC-Undetermined".

Results:

Result	Response	Severity

Usage:

1 Process/Category: Emissions Data Evaluation Report --------- Hg System Integrity Status Evaluation