MOVES NEI Submissions QA Tool

The MOVES National Emission Inventory (NEI) Submissions Quality Assurance (QA) Tool includes scripts to run against onroad and nonroad county database (CDB) submissions for the 2020 NEI. The QA report generated by the QA Tool verifies that all table contents meet naming convention, format, data validity, and other checks. The report confirms that each CDB contains the appropriate tables and that the values within those tables are valid. It is very important that the QA Tool be properly run for each submitted CDB.

The QA Tool is run in MOVES via the command line, using Ant, the MOVES dependency manager and command line interface tool. This document includes specific examples on how to do this. For general information on MOVES' command line tools, see CommandLineMOVES.md.

Prerequisites

In order to run the NEI QA Tool on a county database (CDB), MOVES version 3.0.1 or greater must be installed. Note that MOVES3.0.3 contains minor fixes that improves the tool. The QA Tool also assumes the CDB has already been built using the MOVES County Data Manager or otherwise and is ready to be checked. The tool *will not* aid in creating or populating an NEI CDB.

County Database Naming Convention

To keep track of the thousands of CDBs used in the NEI calculations, EPA has established a naming convention for CDBs that differentiate between databases and make automation of running and processing the inputs and outputs from MOVES easier.

The naming convention for each CDB folder has 20 characters. The first 6 characters identify the county, the next 5 indicate the calendar year of the county database, and the last characters indicate the date on which the database was created.

The first 6 characters consist of the letter `c` followed by the 5-digit Federal Information Processing Standard (FIPS) code for the county, including a leading zero when necessary. The next 5 characters are the letter `y`, followed by a 4-digit calendar year. This calendar year indicates the calendar year of the data contained in the database. A CDB can only contain data from a single calendar year. The last 8 digits, following an underscore character, are the date on which the database was created in a `YYYYMMDD` format.

An example of a CDB name is `c26161y2020_20210601` where this CDB names indicates `c26161` refers to the county FIPS code (in this case Washtenaw County, Michigan). `y2020` refers to the calendar year for the county database and `20210601` identifies the database modification date of June 1, 2021, in `YYYYMMDD` format.

Command Line Interface

To use MOVES' command line tools, first open the Windows command prompt and navigate to the MOVES directory. Then, run the command `setenv`. For more information on using MOVES' command line tools, see CommandLineMOVES.md.

The Ant command to run the NEI QA script is `onroadNEIQA`, and the Ant command to run the nonroad NEI QA script is `nonroadNEIQA`. Inputs to the commands are specified by the following flags:

Flag	Behavior	Examples
`-Dinput`	There are three types of input allowed: 1. A single input database 2. A comma-separated list of databases, wrapped in quotes 3. A .txt file with a list of databases, one on each line	1.`-Dinput=c12345y2020_20210314` 2.`-Dinput="c12345y2020_20210314,c23456y2020_20210314"` 3.`-Dinput=db_list.txt`
`-Doutput`	A file to write the QA results to. Accepted formats are: * `.xlsx` or `.xls` for spreadsheet output * `.csv` for comma-separated output * `.txt` or `.tab` for tab-separated output This input does not need to be wrapped in quotes unless there are spaces or commas in the filename.	1. `-Doutput=PSC_QA_Report.xlsx` 2. `-Doutput="PSC QA Report.csv"` 3. `-Doutput=PSC_QA_Report.txt`

For input option #3 above, databases need to be separated by a new line. The following would be a valid `db_list.txt` file:

```
c12345y2020_20210314
c23456y2020_20210314
```

Putting everything together, the following are all valid ways of calling the QA scripts:

```
setenv
ant onroadNEIQA -Dinput=c12345y2020_20210314 -Doutput=PSC_QA_Report.xlsx
ant onroadNEIQA -Dinput="c12345y2020_20210314,c23456y2020_20210314" -Doutput=PSC_QA_Report.csv
ant onroadNEIQA -Dinput=db_list.txt -Doutput=PSC_QA_Report.txt
ant nonroadNEIQA -Dinput=c12345y2020_20210314 -Doutput=PSC_QA_Report.xlsx
ant nonroadNEIQA -Dinput="c12345y2020_20210314,c23456y2020_20210314" -Doutput=PSC_QA_Report.csv
ant nonroadNEIQA -Dinput=db_list.txt -Doutput="PSC QA Report.txt"
```

Output

The QA scripts create two new tables in the input database, `cdb_checks` and `qa_checks_log`. The contents of `cdb_checks` are copied to the output file. If multiple databases are run using the tool, they are all combined into one output file. `qa_checks_log` is used to log the checks that have been performed, so that users can tell how far the script got if something goes wrong. There are a few additional tables that are only created if the tool detects overlaps and/or gaps in model year ranges in select tables, as described below.

`qa_checks_log` has the following structure:

Field	Type	Description
checkNo	int(11)	check number, which can be mapped to specific checks below
status	char(20)	Status of the check - OK or otherwise
version	char(8)	version of the MOVES default database used as a reference
msgDate	date	date of execution
msgTime	time	time of execution, to the second

`cdb_checks` and the output file have the following structure:

Field	Type	Description
countyID	int(11)	county evaluated in the check
status	char(20)	Status of the check: Complete, Warning, or Error
tableName	char(100)	name of the table evaluated in the check
checkNumber	smallint(6)	number of the check, which can be mapped to specific QA checks in the following sections
testDescription	char(250)	a short description of the check
testValue	text	Additional text providing more context for the failed check
count	int(11)	Number of rows relevant to the check
dataBaseName	char(100)	name of the input database
dayID	smallint(6)	dayID which failed the check, if applicable
fuelFormulationID	smallint(6)	fuelFormulationID which failed the check, if applicable
fuelTypeId	smallint(6)	fuelTypeID which failed the check, if applicable
fuelSubtypeID	smallint(6)	fuelSubTypeID which failed the check, if applicable
fuelYearID	smallint(6)	fuelYearID which failed the check, if applicable
hourDayID	smallint(6)	hourDayID which failed the check, if applicable
hourID	smallint(6)	hourID which failed the check, if applicable
HPMSVtypeID	smallint(6)	HPMSVtypeID which failed the check, if applicable
monthGroupID	smallint(6)	monthGroupID which failed the check, if applicable
monthID	smallint(6)	monthID which failed the check, if applicable
roadTypeID	smallint(6)	roadTypeID which failed the check, if applicable

Field	Type	Description
sourceTypeID	smallint(6)	sourceTypeID which failed the check, if applicable
stateID	smallint(6)	stateID which failed the check, if applicable
yearID	smallint(6)	yearID which failed the check, if applicable
zoneID	int(11)	zoneID which failed the check, if applicable
msgType	char(50)	Type of message: Info, Table Check, or Data Problem
msgDate	date	date of execution
msgTime	time	time of execution, to the second
version	char(8)	version of the MOVES default database used as a reference
sumKeyID	int(11)	
sumKeyDescription	char(50)	

`qa_checks_im` is created if there are overlaps and/or gaps in the `IMCoverage` table's `begModelYearID` and `endModelYearID` columns. It has the following structure:

Field	Type	Description
Cpol	int	polProcessID which failed the check
Ccou	int	countyID which failed the check
Cyea	int	yearID which failed the check
Csou	int	sourceTypeID which failed the check
Cfue	int	fuelTypeID which failed the check
LENDMY	int	the endModelYearID in the row before the row that failed
CBEGMY	int	the begModelYearID which failed the check
CENDMY	int	the endModelYearID which failed the check
useIMyn	char(1)	the value of useIMyn where the check failed
Reason	varchar(40)	a description of why the check failed

`qa_checks_had` is created if there are overlaps and/or gaps in the `hotellingactivitydistribution` table's `beginModelYearID` and `endModelYearID` columns. It has the following structure:

Field	Type	Description
Czone	int	zoneID which failed the check
LENDMY	int	the endModelYearID in the row before the row that failed
CBEGMY	int	the beginModelYearID which failed the check
CENDMY	int	the endModelYearID which failed the check
Reason	varchar(40)	a description of why the check failed

`qa_checks_imyg` is created if there are overlaps and/or gaps in the `idlemodelyeargrouping` table's `minModelYearID` and `maxModelYearID` columns. It has the following structure:

Field	Type	Description
CsourceType	int	sourceTypeID which failed the check
LMaxMY	int	the maxModelYearID in the row before the row that failed
CMinMY	int	the minModelYearID which failed the check
CMaxMY	int	the maxModelYearID which failed the check
Reason	varchar(40)	a description of why the check failed

`qa_checks_tif` is created if there are overlaps and/or gaps in the `totalidlefraction` table's `minModelYearID` and `maxModelYearID` columns. It has the following structure:

Field	Type	Description
CsourceType	int	sourceTypeID which failed the check
Cmonth	int	countyID which failed the check
Cday	int	yearID which failed the check
LMaxMY	int	the maxModelYearID in the row before the row that failed
CMinMY	int	the minModelYearID which failed the check
CMaxMY	int	the maxModelYearID which failed the check
Reason	varchar(40)	a description of why the check failed

Onroad Checks

TableNo	CheckNo	Table	Description	Type	Comment
	1001	hpmsvtypeday or hpmsvtypeyear or sourcetypedayvmt or sourcetypeyearvmt	makes sure VMT is supplied correctly (one and only one of the four possible tables is used)	Error	
	1002	hpmsvtypeday	records the number of rows	Info	
	1003	hpmsvtypeyear	records the number of rows	Info	
	1004	sourcetypedayvmt	records the number of rows	Info	
	1005	sourcetypeyearvmt	records the number of rows	Info	
	1006	starts or startsperday or startsperdaypervehicle	makes sure starts are supplied correctly (either 0 or 1 of the three possible tables are used)	Error	
	1007	starts	records the number of rows	Info	
	1008	startsperday	records the number of rows	Info	
	1009	startsperdaypervehicle	records the number of rows	Info	
11	1100	year	table check (indicates that the checks associated with this table have started)	Info	
	1101	year	checks that isBaseYear is either Y or N	Error	
	1102	year	makes sure fuelYearID is the same as yearID	Error	
	1103	year	checks for unknown yearID	Error	
	1104	year	Checks that the yearID matches the year noted in the CDB name (for example the year '2020' of the CDB `c13121y2020_YYYYMMDD`)	Error	
12	1200	state	table check (indicates that the checks associated with this table have started)	Info	
	1201	state	checks for unknown stateID	Error	
	1202	state	check for unknown idleRegionID	Error	
	1203	state	checks that state has at least 1 row	Error	
13	1300	county	table check (indicates that the checks associated with this table have started)	Info	
	1301	county	checks for unknown countyIDs, referencing the default db	Error	

TableNo	CheckNo	Table	Description	Type	Comment
	1302	county	Checks that the countyID matches the county noted in the CDB name (for example the county '13121' of the CDB `c13121y2020_YYYYMMDD`)	Error	
	1303	county	makes sure altitude column is either L or H	Error	
	1304	county	makes sure the GPAFract is between 0 and 1	Error	
	1305	county	makes sure the barometric pressure is between 20 and 33	Error	
	1306	county	checks stateID against the state table	Error	
	1307	county	checks that county has at least one row	Error	
	1308	county	check for unknown countyTypeID	Error	
14	1400	zone	table check (indicates that the checks associated with this table have started)	Info	
	1401	zone	checks countyID against the county table	Error	
	1402	zone	makes sure startAllocFactor sums to 1	Error	
	1403	zone	makes sure idleAllocFactor sums to 1	Error	
	1404	zone	makes sure SHPAllocFactor sums to 1	Error	
	1405	zone	checks that zoneID is consistent with countyID (zoneID = countyID * 10)	Error	
	1406	zone	checks that zone has at least 1 row	Error	
15	1500	avft	table check (indicates that the checks associated with this table have started)	Info	
	1501	avft	checks for unknown sourceTypeID	Error	
	1502	avft	checks for unknown modelYearID	Error	
	1503	avft	checks for unknown fuelTypeID	Error	
	1504	avft	checks for unknown engTechID	Error	
	1505	avft	makes sure all fuelEngFraction values are between 0 and 1	Error	
	1506	avft	checks that fuelengfraction sums to 1 by source type and model year	Error	

TableNo	CheckNo	Table	Description	Type	Comment
	1507	avft	checks for missing combinations of sourceTypeID, modelYearID, fuelTypeID, and engTechID	Error	This returns only the first missing combination, not all of them
16	1600	avgspeeddistribution	table check (indicates that the checks associated with this table have started)	Info	
	1601	avgspeeddistribution	checks for unknown avgSpeedBinID	Error	
	1602	avgspeeddistribution	checks that the avgSpeedFraction column sums to 1 by sourceTypeID, hourDayID, and roadTypeID	Error	
	1603	avgspeeddistribution	checks for unknown hourDayIDs	Error	
	1604	avgspeeddistribution	checks for unknown roadTypeID	Error	
	1605	avgspeeddistribution	checks for unknown sourceTypeID	Error	
	1606	avgspeeddistribution	checks for missing combinations of hourDayID, roadTypeID, sourceTypeID, and avgSpeedBinID	Error	This returns only the first missing combination, not all of them
	1607	avgspeeddistribution	make sure no avgSpeedFraction values are greater than or equal to 1	Error	
	1608	avgspeeddistribution	make sure no avgSpeedFraction profiles are flat	Warning	For example, all 1/16
	1609	avgspeeddistribution	makes sure the weekday and weekend speed profiles aren't identical	Warning	
	1610	avgspeeddistribution	don't allow identical profiles for each road type	Warning	This checks all pairwise combinations of road types, and will report a warning if any of them match. This will only report the first pair of matching distributions, not all of them
	1611	avgspeeddistribution	checks for 0% avgSpeedFraction in avgSpeedBinID 1	Warning	This returns only the first instance, not all of them
17	1700	countyyear	table check (indicates that the checks associated with this table have started)	Info	
	1701	countyyear	checks countyID against the county table	Error	
	1702	countyyear	checks yearID against year table	Error	
	1703	countyyear	checks that refuelingVaporProgramAdjustment is between 0 and 1	Error	
	1704	countyyear	checks that refuelingSpillProgramAdjustment is between 0 and 1	Error	

TableNo	CheckNo	Table	Description	Type	Comment
18	1800	dayvmtfraction	table check (indicates that the checks associated with this table have started)	Info	
	1801	dayvmtfraction	checks for unknown dayID	Error	
	1802	dayvmtfraction	makes sure dayVMTFraction sums to 1 by monthID, roadTypeID, and sourceTypeID	Error	
	1803	dayvmtfraction	checks for unknown monthID	Error	
	1804	dayvmtfraction	checks for uknown roadTypeID	Error	
	1805	dayvmtfraction	checks for unknown sourceTypeID	Error	
	1806	dayvmtfraction	make sure no dayVMTFraction values are greater than or equal to 1	Error	
	1807	dayvmtfraction	makes sure allocations between weekend and weekday are not identical	Warning	
	1808	dayvmtfraction	If populated, makes sure there are no missing combinations of monthID, roadTypeID, sourceTypeID, and dayID	Error	This returns only the first missing combination, not all of them
19	1900	emissionratebyage	table check (indicates that the checks associated with this table have started)	Info	
	1901	emissionratebyage	checks for unknown polProcessID	Error	
	1902	emissionratebyage	checks for unknown opModeID	Error	
	1903	emissionratebyage	checks for unknown ageGroupID	Error	
20	2000	fuelformulation	table check (indicates that the checks associated with this table have started)	Info	
	2001	fuelformulation	checks for uknown fuelSubtypeID	Warning	
	2002	fuelformulation	checks for valid RVP (>=5 && <=20) for all gasoline fuelSubTypeIDs, not including E85	Warning	
	2003	fuelformulation	checks for valid sulfur levels (0-5000) for all fuels	Warning	
	2004	fuelformulation	checks for valid ethanol volume (0-100) for user-supplied fuels (fuelFormulationID > 100)	Warning	
	2005	fuelformulation	checks that MTBE is 0 or NULL for user-supplied fuels (fuelFormulationID > 100)	Warning	
	2006	fuelformulation	checks that ETBE is 0 or NULL for user-supplied fuels (fuelFormulationID > 100)	Warning	

TableNo	CheckNo	Table	Description	Type	Comment
	2007	fuelformulation	checks that TAME is 0 or NULL for user-supplied fuels (fuelFormulationID > 100)	Warning	
	2008	fuelformulation	checks for valid aromatic content (0-55) for user-supplied gasoline fuelSubTypeIDs (fuelFormulationID > 100)	Warning	
	2009	fuelformulation	checks for valid olefin content (0-25) for user-supplied gasoline fuelSubTypeIDs (fuelFormulationID > 100)	Warning	
	2010	fuelformulation	checks for valid benzene content (0-5) for user-supplied gasoline fuelSubTypeIDs (fuelFormulationID > 100)	Warning	
	2011	fuelformulation	checks for valid e200 values (0-70) for user-supplied gasoline fuelSubTypeIDs (fuelFormulationID > 100)	Warning	
	2012	fuelformulation	checks for valid e300 values (0-100) for user-supplied gasoline fuelSubTypeIDs (fuelFormulationID > 100)	Warning	
	2013	fuelformulation	makes sure the T50 and T90 columns exist	Error	
21	2100	fuelsupply	table check (indicates that the checks associated with this table have started)	Info	
	2101	fuelsupply	checks for unknown fuelFormulationIDs, referencing the CDB fuelforumulation table	Warning	
	2102	fuelsupply	checks fuelYearID against the year table	Warning	
	2103	fuelsupply	checks for unknown monthGroupID	Warning	
	2104	fuelsupply	checks for multiple fuelRegionIDs	Warning	
22	2200	fuelusagefraction	table check (indicates that the checks associated with this table have started)	Info	
	2201	fuelusagefraction	checks fuelYearID against the year table	Error	
	2202	fuelusagefraction	checks countyID against the county table	Error	
	2203	fuelusagefraction	checks for unknown modelYearGroupID	Error	
	2204	fuelusagefraction	checks for unknown sourceBinFuelTypeID	Error	
	2205	fuelusagefraction	checks for unknown fuelSupplyFuelTypeID	Error	

TableNo	CheckNo	Table	Description	Type	Comment
	2206	fuelusagefraction	checks that sourceBinFuelTypeID = fuelSupplyFuelTypeID for non FF vehicles	Error	
	2207	fuelusagefraction	checks that fuelSupplyFuelTypeID is 1 or 5 for FF vehicles	Error	
	2208	fuelusagefraction	make sure theres a row for sourceBinFuelTypeID 5 where fuelSupplyFuelTypeID is 1 or 5	Error	
	2209	fuelusagefraction	make sure usageFraction sums to 1 for all sourceBinFuelTypeIDs	Error	
23	2300	hotellingactivitydistribution	table check (indicates that the checks associated with this table have started)	Info	
	2301	hotellingactivitydistribution	checks for unknown opModeID	Error	
	2302	hotellingactivitydistribution	checks that opModeDistribution sums to 1 by model year group	Error	
	2303	hotellingactivitydistribution	check zoneID against zone table	Error	
	2304	hotellingactivitydistribution	check for overlaps/gaps in model year columns	Error	
24	2400	hotellingagefraction	table check (indicates that the checks associated with this table have started)	Info	
	2401	hotellingagefraction	checks zoneID against zone table	Error	
	2402	hotellingagefraction	checks for unknown ageID	Error	
	2403	hotellingagefraction	If populated, makes sure all ages are present	Error	This returns only the first missing ageID, not all of them
	2404	hotellingagefraction	makes sure no ageFraction values are greater than or equal to 1	Error	
	2405	hotellingagefraction	make sure ageFraction sums to 1 by zoneID	Error	
	2406	hotellingagefraction	make sure no fraction profiles are flat	Warning	For example, all 1/31
25	2500	hotellinghourfraction	table check (indicates that the checks associated with this table have started)	Info	
	2501	hotellinghourfraction	check zoneID against zone table	Error	
	2502	hotellinghourfraction	check for unknown dayID	Error	
	2503	hotellinghourfraction	check for unknown hourID	Error	
	2504	hotellinghourfraction	If populated, makes sure all combinations of zoneID, dayID, and hourID are present	Error	This returns only the first missing combination, not all of them

TableNo	CheckNo	Table	Description	Type	Comment
	2505	hotellinghourfraction	make sure no hourFraction values are greater than or equal to 1	Error	
	2506	hotellinghourfraction	make sure all hourFractions sum to 1 by zoneID and dayID	Error	
	2507	hotellinghourfraction	make sure no hour profiles are flat	Warning	For example, all 1/24
	2508	hotellinghourfraction	makes sure hour profiles aren't identical for weekends and weekdays	Warning	
26	2600	hotellinghoursperday	table check (indicates that the checks associated with this table have started)	Info	
	2601	hotellinghoursperday	check yearID against year table	Error	
	2602	hotellinghoursperday	check zoneID against zone table	Error	
	2603	hotellinghoursperday	check for unknown dayID	Error	
	2604	hotellinghoursperday	If populated, makes sure all combinations of yearID, zoneID, and dayID are present	Error	This returns only the first missing combination, not all of them
27	2700	hotellingmonthadjust	table check (indicates that the checks associated with this table have started)	Info	
	2701	hotellingmonthadjust	Check zoneID against zone table	Error	
	2702	hotellingmonthadjust	Check for unknown monthID	Error	
	2703	hotellingmonthadjust	If populated, makes sure all combinations of zoneID and monthID are present	Error	This returns only the first missing combination, not all of them
28	2800	hourvmtfraction	table check (indicates that the checks associated with this table have started)	Info	
	2801	hourvmtfraction	checks for unknown dayID	Error	
	2802	hourvmtfraction	checks for unknown hourID	Error	
	2803	hourvmtfraction	checks for unknown roadTypeID	Error	
	2804	hourvmtfraction	checks for unknown sourceTypeID	Error	
	2805	hourvmtfraction	If populated, makes sure hourVMTFraction sums to 1 for each dayID, roadTypeID, and sourceTypeID	Error	This returns only the first missing combination, not all of them
	2806	hourvmtfraction	makes sure all combinations of dayID, hourID, roadTypeID, and sourceTypeID are present	Error	

TableNo	CheckNo	Table	Description	Type	Comment
	2807	hourvmtfraction	make sure no hourVMTFraction values are greater than or equal to 1	Error	
	2808	hourvmtfraction	make sure no hour profiles are flat	Warning	For example, all 1/24
	2809	hourvmtfraction	makes sure hour profiles aren't identical for weekends and weekdays	Warning	
29	2900	hpmsvtypeday	table check (indicates that the checks associated with this table have started)	Info	
	2901	hpmsvtypeday	checks yearID against year table	Error	
	2902	hpmsvtypeday	checks for unknown monthID	Error	
	2903	hpmsvtypeday	checks for unknown dayID	Error	
	2904	hpmsvtypeday	check for unknown hpmsvtypeID	Error	
	2905	hpmsvtypeday	If populated, makes sure all combinations of yearID, monthID, dayID, and HPMSVtypeID are present	Error	This returns only the first missing combination, not all of them
30	3000	hpmsvtypeyear	table check (indicates that the checks associated with this table have started)	Info	
	3001	hpmsvtypeyear	checks for unknown HPMSVtypeID	Error	
	3002	hpmsvtypeyear	checks for unknown yearID	Error	
	3003	hpmsvtypeyear	If populated, makes sure all combinations of yearID and HPMSVtypeID are present	Error	This returns only the first missing combination, not all of them
31	3100	idledayadjust	table check (indicates that the checks associated with this table have started)	Info	
	3101	idledayadjust	Check for unknown sourceTypeID	Error	
	3102	idledayadjust	Check for unknown dayID	Error	
	3103	idledayadjust	If populated, make sure all combinations of sourceTypeID and dayID are present	Error	This returns only the first missing combination, not all of them
32	3200	idlemodelyeargrouping	table check (indicates that the checks associated with this table have started)	Info	
	3201	idlemodelyeargrouping	Check for unknown sourceTypeID	Error	
	3202	idlemodelyeargrouping	If populated, makes sure all sourceTypeIDs are present	Error	This returns only the first missing sourceTypeID, not all of them
	3203	idlemodelyeargrouping	Check for model year gaps and overlaps	Error	

TableNo	CheckNo	Table	Description	Type	Comment
33	3300	idlemonthadjust	table check (indicates that the checks associated with this table have started)	Info	
	3301	idlemonthadjust	Check for unknown sourceTypeID	Error	
	3302	idlemonthadjust	Check for unknown monthID	Error	
	3303	idlemonthadjust	If populated, makes sure all combinations of sourceTypeID and monthID are present	Error	This returns only the first missing combination, not all of them
34	3400	totalidlefraction	table check (indicates that the checks associated with this table have started)	Info	
	3401	totalidlefraction	Check for unknown sourceTypeID	Error	
	3402	totalidlefraction	Check for unknown monthID	Error	
	3403	totalidlefraction	Check for unknown dayID	Error	
	3404	totalidlefraction	Check for unknown idleRegionID	Error	
	3405	totalidlefraction	checks for multiple idleRegionIDs	Error	
	3406	totalidlefraction	Check for unknown countyTypeID	Error	
	3407	totalidlefraction	checks for multiple countyTypeIDs	Error	
	3408	totalidlefraction	Check for model year gaps and overlaps	Error	
	3409	totalidlefraction	If populated, checks that all combinations of sourceTypeID, monthID, dayID, idleRegionID, and countyTypeID are present	Error	This returns only the first missing combination, not all of them
35	3500	imcoverage	table check (indicates that the checks associated with this table have started)	Info	
	3501	imcoverage	checks countyID against the county table	Error	
	3502	imcoverage	checks for unknown fuelTypeID	Error	
	3503	imcoverage	checks for unknown polProcessID	Error	
	3504	imcoverage	checks that inspection frequency is either 1 or 2	Error	
	3505	imcoverage	checks that useIMyn is either "y" or "n", case insensitive	Error	
	3506	imcoverage	checks that complianceFactor is between 0 and 100	Error	
	3507	imcoverage	checks for unknown sourceTypeID	Error	
	3508	imcoverage	checks stateID against the state table	Error	

TableNo	CheckNo	Table	Description	Type	Comment
	3509	imcoverage	checks yearID against year table	Error	
	3510	imcoverage	makes sure fuel types 1 and 5 have the same number of rows	Error	
	3511	imcoverage	checks for model year overlaps and gaps	Error	
36	3600	monthvmtfraction	table check (indicates that the checks associated with this table have started)	Info	
	3601	monthvmtfraction	checks for unknown monthID	Error	
	3602	monthvmtfraction	checks for unknown sourceTypeID	Error	
	3603	monthvmtfraction	makes sure monthvmtfraction sums to 1 for each sourceTypeID	Error	
	3604	monthvmtfraction	make sure no monthVMTFraction values are greater than or equal to 1	Error	
	3605	monthvmtfraction	make sure no month profiles are flat	Warning	For example, all 1/12
	3606	monthvmtfraction	If populated, makes sure all combinations of sourceTypeID and monthID are present	Error	This returns only the first missing combination, not all of them
37	3700	onroadretrofit	table check (indicates that the checks associated with this table have started)	Info	
	3701	onroadretrofit	checks for unknown pollutantID	Error	
	3702	onroadretrofit	checks for unknown processID	Error	
	3703	onroadretrofit	checks for unknown fuelTypeID	Error	
	3704	onroadretrofit	checks for unknown sourceTypeID	Error	
	3705	onroadretrofit	checks that retrofitYearID is less than or equal to analysis year	Error	
	3706	onroadretrofit	checks that endModelYearID is less than or equal to retrofitYearID	Error	
	3707	onroadretrofit	checks that beginModelYearID is less than or equal to endModelYearID	Error	
	3708	onroadretrofit	checks that cumFractionRetrofit is between 0 and 1	Error	
	3709	onroadretrofit	checks that retrofitEffectiveFraction is less than or equal to 1	Error	

TableNo	CheckNo	Table	Description	Type	Comment
38	3800	roadtypedistribution	table check (indicates that the checks associated with this table have started)	Info	
	3801	roadtypedistribution	checks for unknown roadTypeID	Error	
	3802	roadtypedistribution	checks for unknown sourceTypeID	Error	
	3803	roadtypedistribution	checks that VMTFraction sums to 1 by sourceTypeID	Error	
	3804	roadtypedistribution	make sure no VMTFraction values are greater than or equal to 1	Warning	
	3805	roadtypedistribution	make sure no road type profiles are flat	Warning	For example, all 1/4
	3806	roadtypedistribution	makes sure all combinations of roadTypeID and sourceTypeID are present	Error	This returns only the first missing combination, not all of them
39	3900	sourcetypeagedistribution	table check (indicates that the checks associated with this table have started)	Info	
	3901	sourcetypeagedistribution	checks for unknown ageID	Error	
	3902	sourcetypeagedistribution	checks for unknown sourceTypeID	Error	
	3903	sourcetypeagedistribution	checks yearID against year table	Error	
	3904	sourcetypeagedistribution	checks that sourceTypeAgeFraction sums to 1 for each ageID, sourceTypeID, and yearID	Error	
	3905	sourcetypeagedistribution	checks for missing sourceTypeID, ageID, yearID combinations	Error	This returns only the first missing combination, not all of them
	3906	sourcetypeagedistribution	make sure no age profiles are flat	Warning	For example, all 1/31
40	4000	sourcetypedayvmt	table check (indicates that the checks associated with this table have started)	Info	
	4001	sourcetypedayvmt	checks yearID against year table	Error	
	4002	sourcetypedayvmt	checks for unknown monthID	Error	
	4003	sourcetypedayvmt	checks for unknown dayID	Error	
	4004	sourcetypedayvmt	checks for unknown sourceTypeID	Error	
	4005	sourcetypedayvmt	If populated, makes sure all yearID, monthID, dayID, and sourceTypeID combinations are present	Error	This returns only the first missing combination, not all of them
41	4100	sourcetypeyearvmt	table check (indicates that the checks associated with this table have started)	Info	

TableNo	CheckNo	Table	Description	Type	Comment
	4101	sourcetypeyearvmt	checks yearID against year table	Error	
	4102	sourcetypeyearvmt	checks for unknown sourceTypeID	Error	
	4103	sourcetypeyearvmt	If populated, makes sure all combinations of yearID and sourceTypeID are present	Error	This returns only the first missing combination, not all of them
42	4200	sourcetypeyear	table check (indicates that the checks associated with this table have started)	Info	
	4201	sourcetypeyear	checks yearID against year table	Error	
	4202	sourcetypeyear	checks for unknown sourceTypeID	Error	
	4203	sourcetypeyear	makes sure all combinations of yearID and sourceTypeID are present	Error	This returns only the first missing combination, not all of them
	4204	sourcetypeyear or HPMSVtypeDay	checks that population exists for HPMS types with VMT and that no population exists for HPMS types without VMT, if using hpmsvtypeday for VMT	Error	
	4205	sourcetypeyear or HPMSVtypeYear	checks that population exists for HPMS types with VMT and that no population exists for HPMS types without VMT, if using hpmsvtypeyear for VMT	Error	
	4206	sourcetypeyear or sourceTypeDayVMT	checks that population exists for source types with VMT and that no population exists for source types without VMT, if using sourcetypedayvmt for VMT	Error	
	4207	sourcetypeyear or sourceTypeYearVMT	checks that population exists for source types with VMT and that no population exists for source types without VMT, if using sourcetypeyearvmt for VMT	Error	
43	4300	starts	table check (indicates that the checks associated with this table have started)	Info	
	4301	starts	checks for unknown hourDayID	Error	
	4302	starts	checks for unknown monthID	Error	
	4303	starts	checks yearID against year table	Error	
	4304	starts	checks for unknown ageID	Error	
	4305	starts	checks zoneID against zone table	Error	
	4306	starts	checks for unknown sourceTypeID	Error	

TableNo	CheckNo	Table	Description	Type	Comment
	4307	starts	check isUserInput is Y, y, N, or n	Error	
	4308	starts	If populated, checks for all combinations of hourDayID, monthID, yearID, ageID, zoneID, and sourceTypeID	Error	This returns only the first missing combination, not all of them
44	4400	startsagadjustment	table check (indicates that the checks associated with this table have started)	Info	
	4401	startsagadjustment	Check for unknown sourceTypeID	Error	
	4402	startsagadjustment	Check for unknown ageID	Error	
	4403	startsagadjustment	If populated, makes sure all sourceTypeID, ageID combinations are present	Error	This returns only the first missing combination, not all of them
45	4500	startshourfraction	table check (indicates that the checks associated with this table have started)	Info	
	4501	startshourfraction	checks for unknown dayID	Error	
	4502	startshourfraction	checks for unknown hourID	Error	
	4503	startshourfraction	check for unknown sourceTypeID	Error	
	4504	startshourfraction	makes sure allocationFraction sums to 1 by zoneID and dayID	Error	
	4505	startshourfraction	make sure no allocationFraction values are greater than or equal to 1	Error	
	4506	startshourfraction	make sure no hour profiles are flat	Warning	For example, all 1/24
	4507	startshourfraction	If populated, makes sure all combinations of dayID, sourceTypeID, and hourID are present	Error	This returns only the first missing combination, not all of them
46	4600	startsmonthadjust	table check (indicates that the checks associated with this table have started)	Info	
	4601	startsmonthadjust	checks for unknown monthID	Error	
	4602	startsmonthadjust	check for unknown sourceTypeID	Error	
	4603	startsmonthadjust	If populated, makes sure all combinations of monthID and sourceTypeID are present	Error	This returns only the first missing combination, not all of them
47	4700	startsopmodedistribution	table check (indicates that the checks associated with this table have started)	Info	
	4701	startsopmodedistribution	Check for unknown dayID	Error	
	4702	startsopmodedistribution	Check for unknown hourID	Error	

TableNo	CheckNo	Table	Description	Type	Comment
	4703	startsopmodedistribution	Check for unknown sourceTypeID	Error	
	4704	startsopmodedistribution	Check for unknown ageID	Error	
	4705	startsopmodedistribution	Check for unknown opModeID	Error	
	4706	startsopmodedistribution	Check isUserInput is 'Y' or 'N', case insensitive	Error	
	4707	startsopmodedistribution	Make sure opModeFraction sums to 1 by dayID, hourID, sourceTypeID, and ageID	Error	
	4708	startsopmodedistribution	If populated, makes sure all combinations of dayID, hourID, sourceTypeID, ageID, and opModeID are present	Error	This returns only the first missing combination, not all of them
48	4800	startsperday	table check (indicates that the checks associated with this table have started)	Info	
	4801	startsperday	checks for unknown dayID	Error	
	4802	startsperday	check for unknown sourceTypeID	Error	
	4803	startsperday	If populated, makes sure all combinations of dayID and sourceTypeID are present	Error	This returns only the first missing combination, not all of them
49	4900	startsperdaypervehicle	table check (indicates that the checks associated with this table have started)	Info	
	4901	startsperdaypervehicle	Check for unknown dayID	Error	
	4902	startsperdaypervehicle	Check for unknown sourceTypeID	Error	
	4903	startsperdaypervehicle	If populated, makes sure all combinations of dayID and sourceTypeID are present	Error	This returns only the first missing combination, not all of them
50	5000	zonemonthhour	table check (indicates that the checks associated with this table have started)	Info	
	5001	zonemonthhour	checks for unknown hourID	Warning	
	5002	zonemonthhour	checks for unkown monthID	Warning	
	5003	zonemonthhour	checks zoneID against zone table	Warning	
	5004	zonemonthhour	makes sure temperature is between -80F and 150 F	Warning	
	5005	zonemonthhour	makes sure relative humidity is between 0 and 100	Warning	
	5006	zonemonthhour	If populated, make sure all combinations of zoneID, monthID, and hourID are populated	Warning	This returns only the first missing combination, not all of them

TableNo	CheckNo	Table	Description	Type	Comment
51	5100	zoneroadtype	table check (indicates that the checks associated with this table have started)	Info	
	5101	zoneroadtype	checks for unknown roadTypeID	Error	
	5102	zoneroadtype	checks zoneID against zone table	Error	
	5103	zoneroadtype	makes sure SHOAllocFactor sums to 1 for each roadTypeID, and there is only one row per road type	Error	
	5104	zoneroadtype	Makes sure all combinations of roadTypeID and zoneID are present	Error	This returns only the first missing combination, not all of them

Nonroad Checks

TableNo	CheckNo	Table	Description	Type
11	1	nrbaseyearequippopulation	checks if table is missing or empty	Info if missing, warning if empty
	21	nrbaseyearequippopulation	checks for unknown sourceTypeID	Error
	22	nrbaseyearequippopulation	checks for unknown stateID (using default db)	Error
13	3	nrmonthallocation	checks if table is missing or empty	Info if missing, warning if empty
	23	nrmonthallocation	checks for unknown stateID (using default db)	Error
	24	nrmonthallocation	checks for unknown monthID	Error
	25	nrmonthallocation	makes sure monthFraction sums to 1 where appropriate	Error
14	4	nrdayallocation	checks if table is missing or empty	Info if missing, warning if empty
	26	nrdayallocation	checks for unknown dayID	Error
	27	nrdayallocation	makes sure dayfraction sums to 1	Error
15	5	nrhourallocation	checks if table is missing or empty	Info if missing, warning if empty
	28	nrhourallocation	checks for unknown hourID, hardcoded to be between 1 and 24	Error
	29	nrhourallocation	makes sure hourfraction sums to 1	Error
16	6	nrretrofitfactors	checks if table is missing or empty	Info if missing, warning if empty
	3701	nrretrofitfactors	checks for unknown pollutantIDs	Error
	3703	nrretrofitfactors	check that the retrofitStartYear <= retrofitEndYear	Error
	3704	nrretrofitfactors	check that EndModelYear <= retrofitEndYear	Error

TableNo	CheckNo	Table	Description	Type
	3705	nrretrofitfactors	check that StartModelYear <= EndModelYear	Error
	3706	nrretrofitfactors	check that annualFractionRetrofit between 0 and 1	Error
	3707	nrretrofitfactors	check that retrofitEffectiveFraction <= 1	Error
17	7	nrsourceusetype	checks if table is missing or empty	Info if missing, warning if empty
18	8	nrstatesurrogate	checks if table is missing or empty	Info if missing, warning if empty
	33	nrstatesurrogate	checks for unknown stateID (using default db)	Error
	34	nrstatesurrogate	checks for unknown countyID (using default db)	Error
19	9	fuelformulation	checks if table is missing or empty	Info if missing, warning if not empty
	51	fuelformulation	checks fror unknown fuelsubtypeid	Error
	52	fuelformulation	checks for RVP between 5 and 20 for gasoline fuel subtypes	Error
	53	fuelformulation	checks that sulfur is between 0 and 5000	Error
	54	fuelformulation	checks that ETOHVolume is between 0 and 100 for gasoline fuel subtypes	Error
	55	fuelformulation	checks for valid MTBE (0 or NULL)	Error
	56	fuelformulation	checks for valid ETBE (0 or NULL)	Error
	57	fuelformulation	checks for valid TAME (0 or NULL)	Error
	58	fuelformulation	checks for valid aromatic content (0-55) for gasoline fuelSubTypeIDs	Error

TableNo	CheckNo	Table	Description	Type
	59	fuelformulation	checks for valid olefin content (0-25) for gasoline fuelSubTypes	Error
	60	fuelformulation	checks for valid benzene content (0-5) for gasoline fuelSubTypes	Error
	61	fuelformulation	checks for valid e200 values (0-70) for gasoline fuelSubTypes	Error
	62	fuelformulation	checks for valid e300 values (0-100) for gasoline fuelSubTypes	Error
	64	fuelformulation	checks that the T50 and T90 columns exist	Error
20	10	nrfuelsupply	checks if table is missing or empty	Info if missing, warning if not empty
	71	nrfuelsupply	checks for unknown fuelRegionID	Error
	72	nrfuelsupply	checks for unknown fuelFormulationID	Error
	73	nrfuelsupply	checks for unknown fuelYearID (using default db)	Error
	74	nrfuelsupply	checks that marketShare sums to 1	Error
	75	nrfuelsupply	checks for unknown monthGroupID	Error
21	11	zonemonthhour	checks if table is missing or empty	Info if missing, warning if not empty
	81	zonemonthhour	checks for unknown hourID	Error
	82	zonemonthhour	checks for unkown monthID	Error
	83	zonemonthhour	makes sure temperature is between -80F and 150 F	Error
	84	zonemonthhour	makes sure humidity is between 0 and 100	Error
	85	zonemonthhour	checks for unknown zoneID	Error

TableNo	CheckNo	Table	Description	Type
22	12	nrgrowthindex	checks if table is missing or empty	Info if missing, warning if empty