# **Event Date Related Validation Tests**

Following are listed the event date-related validation tests. The characteristics of each test are given, including the link to the full description and comments in GitHub, the conditions for each expected response, special notes about the test, the number and percent of total occurrences for each category of expected response, and the SQL statements used to get the counts.

Test: TG2\_VALIDATION\_EVENT\_EMPTY

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/88>

**Expected response**: COMPLIANT if at least one field needed to determine the event date

exists and is not EMPTY; otherwise NOT\_COMPLIANT.

**Notes**: Only fails if all of the relevant fields of the Darwin Core Event class are EMPTY

or do not exist. Relevant Darwin Core fields include eventID, eventDate,

verbatimEventDate, year, month, day, startDayOfYear, endDayOfYear.

**NOT\_COMPLIANT**: 70,489,570 (5.8%)

    SELECT sum(occcount) as occcounts FROM `gbif\_dates`

    WHERE

    v\_eventdate IS NULL AND

    v\_year IS NULL AND

    v\_month IS NULL AND

    v\_day IS NULL AND

    v\_year IS NULL AND

    v\_verbatimeventdate IS NULL AND

    v\_startdayofyear IS NULL AND

    v\_enddayofyear IS NULL

**COMPLIANT**: 1,142,920,425 (94.2%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE

    v\_eventdate IS NOT NULL OR

    v\_year IS NOT NULL OR

    v\_month IS NOT NULL OR

    v\_day IS NOT NULL OR

    v\_year IS NOT NULL OR

    v\_verbatimeventdate IS NOT NULL OR

    v\_startdayofyear IS NOT NULL OR

    v\_enddayofyear IS NOT NULL

Test: TG2\_VALIDATION\_EVENTDATE\_EMPTY

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/33>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if the field dwc:eventDate is not present; COMPLIANT if the value of the field dwc:eventDate is not EMPTY; otherwise NOT\_COMPLIANT

**Notes**: None

**INTERNAL\_PREREQUISITES\_NOT\_MET** or **NOT\_COMPLIANT**: 672,432,104 (55.4%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_eventdate IS NULL

**COMPLIANT**: 540,977,891 (44.6%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_eventdate IS NOT NULL

**Comment**: It was not possible to distinguish between results from the categories INTERNAL\_PREREQUISITES\_NOT\_MET and NOT\_COMPLIANT, because in the GBIF snapshot we did not have access to the original data, so we were unable to determine if an EMPTY value was due to a missing value or a missing field.

Test: TG2\_VALIDATION\_EVENTDATE\_NOTSTANDARD

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/66>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if the field dwc:eventDate is either not present or is EMPTY; COMPLIANT if the value of dwc:eventDate is a valid ISO 8601-1:2019 date; otherwise NOT\_COMPLIANT

**Notes**: This test should also pick up issues such as 29 Feb in a non-leap year.

**INTERNAL\_PREREQUISITES\_NOT\_MET**: 672,432,104 (55.4%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_eventdate IS NULL

**NOT\_COMPLIANT**: 38,669,033 (3.6%, estimated)

**COMPLIANT**: 502,308,858 (41.4%, estimated)

SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

WHERE v\_eventdate IS NOT NULL

AND regexp\_contains(v\_eventdate,r'^([\+-]?\d{4})((-?)((0[1-9]|1[0-2])((-?)([12]\d|0[1-9]|3[01]))?|W([0-4]\d|5[0-2])(-?[1-7])?|(00[1-9]|0[1-9]\d|[12]\d{2}|3([0-5]\d|6[1-6])))([T\s]((([01]\d|2[0-3])((:?)[0-5]\d)?|24\:?00)([\.,]\d+)?)?((:?)[0-5]\d([\.,]\d+)?)?([zZ]|([\+-])([01]\d|2[0-3]):?([0-5]\d)?)?)?)?((\/?)([\+-]?\d{4})((-?)((0[1-9]|1[0-2])((-?)([12]\d|0[1-9]|3[01]))?|W([0-4]\d|5[0-2])(-?[1-7])?|(00[1-9]|0[1-9]\d|[12]\d{2}|3([0-5]\d|6[1-6])))([T\s]((([01]\d|2[0-3])((:?)[0-5]\d)?|24\:?00)([\.,]\d+)?)?((:?)[0-5]\d([\.,]\d+)?)?([zZ]|([\+-])([01]\d|2[0-3]):?([0-5]\d)?)?)?))?$') = true

Regex adapted from <https://gist.github.com/philipashlock/8830168> to match compliant single date or date range.

Test: TG2\_VALIDATION\_EVENTDATE\_OUTOFRANGE

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/36>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if dwc:eventDate is either not present or is EMPTY; COMPLIANT if the no part of the range of dwc:eventDate extends outside

optionally-provided begin and end dates; otherwise NOT\_COMPLIANT. If no end date is

provided, the test should use the current time as an upper bound.

**Notes**: The results of this test are time-dependent. Today the date for tomorrow is not

valid. Tomorrow it will be. This test provides the option to designate lower and upper

limits to the date. The upper limit, if not provided should default to the time when the

test is run. There should be no default lower limit. NB By convention, use 1700-01-01 as

a lower limit for collecting dates of biological specimens.

**INTERNAL\_PREREQUISITES\_NOT\_MET**: 672,432,104 (55.4%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_eventdate IS NULL

**NOT\_COMPLIANT**: ?

**COMPLIANT**: ?

**Comment:** The queries for compliance were not constructed for this test.

Test: TG2\_VALIDATION\_EVENTDATE\_INCONSISTENT

**GitHub Issue:**<https://github.com/tdwg/bdq/issues/67>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if dwc:eventDate is either not present or is EMPTY, or all of dwc:year, dwc:month, dwc:day, dwc:startDayOfYear and

dwc:endDayOfYear are not present or are EMPTY; COMPLIANT if the provided values for

dwc:year, dwc:month, dwc:day, dwc:startDayOfYear and dwc:endDayofYear are within the

range of the supplied dwc:eventDate; otherwise NOT\_COMPLIANT

**Notes**: It is expected that amendments on the event date fields are attempted before this

test is run. For this test, dwc:eventTime is explicitly ignored. It may be useful to

consider an additional test that does evaluate dwc:eventTime and dwc:eventDate. In that

case, but not in this test, if the time is present in both dwc:eventDate and

dwc:eventTime, and it is inconsistent, it may indicate an error in the dwc:eventDate,

thus making it a problem that someone needs to evaluate.

**INTERNAL\_PREREQUISITES\_NOT\_MET**: 935,227,465 (77.1%)

SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

WHERE v\_eventdate IS NULL

OR (

v\_year IS NULL AND

v\_month IS NULL AND

v\_day IS NULL AND

v\_startdayofyear IS NULL AND

v\_enddayofyear IS NULL)

**NOT\_COMPLIANT**: ?

**COMPLIANT**: ?

**Comment:** The queries for compliance were not constructed for this test.

Test: TG2\_VALIDATION\_YEAR\_EMPTY (v\_year)

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/49>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if the field dwc:year is not present; COMPLIANT if the value of the field dwc:year is not EMPTY; otherwise NOT\_COMPLIANT

**Notes**: None

**INTERNAL\_PREREQUISITES\_NOT\_MET** or **NOT\_COMPLIANT**: 343,514,283 (28.3%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_year IS NULL

**COMPLIANT**: 869,895,712 (71.7%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_year IS NOT NULL

**Comment**: It was not possible to distinguish between results from the categories INTERNAL\_PREREQUISITES\_NOT\_MET and NOT\_COMPLIANT, because in the GBIF snapshot we did not have access to the original data, so we were unable to determine if an EMPTY value was due to a missing value or a missing field.

Test: TG2\_VALIDATION\_YEAR\_EMPTY (year)

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/49>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if the field dwc:year is not present; COMPLIANT if the value of the field dwc:year is not EMPTY; otherwise NOT\_COMPLIANT

**Notes**: None

**INTERNAL\_PREREQUISITES\_NOT\_MET** or **NOT\_COMPLIANT**: 92,390,420 (7.6%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE year IS NULL

**COMPLIANT**: 1,121,019,575 (92.4.%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE year IS NOT NULL

**Comment**: It was not possible to distinguish between results from the categories INTERNAL\_PREREQUISITES\_NOT\_MET and NOT\_COMPLIANT, because in the GBIF snapshot we did not have access to the original data, so we were unable to determine if an EMPTY value was due to a missing value or a missing field.

Test: TG2\_VALIDATION\_YEAR\_NOTSTANDARD (v\_year)

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/141>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if dwc:year is not present, or is EMPTY or can not be cast as an integer; COMPLIANT if the value of dwc:year cast as an integer does not extend outside optionally-provided begin and end years; otherwise NOT\_COMPLIANT

**Notes**: The results of this test are time-dependent. Next year is not valid now; next year it will be. This test provides the option to designate lower and upper limits to the year. The upper limit, if not provided, should default to the year when the test is run. NB By convention, use 1600 as a lower limit for collecting dates of biological specimens.

**Parameter(s)**: Default values: earliest year = 1600, latest year = current year

**INTERNAL\_PREREQUISITES\_NOT\_MET**: 343,514,283 (28.3%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_year IS NULL

**NOT\_COMPLIANT**: 2,554,982 (0.2%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_year IS NOT NULL

    AND (SAFE\_CAST(v\_year AS INT64)>2019

    OR SAFE\_CAST(v\_year AS INT64)<1600

    OR SAFE\_CAST(v\_year AS INT64) IS NULL)

**COMPLIANT**: 867,340,730 (71.5%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE SAFE\_CAST(v\_year AS INT64)<=2019

    AND SAFE\_CAST(v\_year AS INT64)>=1600

Test: TG2\_VALIDATION\_YEAR\_NOTSTANDARD (year)

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/141>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if dwc:year is not present, or is EMPTY or can not be cast as an integer; COMPLIANT if the value of dwc:year cast as an integer does not extend outside optionally-provided begin and end years; otherwise NOT\_COMPLIANT

**Notes**: The results of this test are time-dependent. Next year is not valid now;. nNext year it will be. This test provides the option to designate lower and upper limits to the year. The upper limit, if not provided, should default to the year when the test is run. NB By convention, use 1600 as a lower limit for collecting dates of biological specimens.

**Parameter(s)**: Default values: earliest year = 1600, latest year = current year

**INTERNAL\_PREREQUISITES\_NOT\_MET**: 92,390,420 (7.6%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE year IS NULL

**NOT\_COMPLIANT**: 0 (0%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE year NOT BETWEEN 1600 AND 2019

**COMPLIANT**: 1,121,019,575 (92.4%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE year BETWEEN 1600 AND 2019

Test: TG2\_VALIDATION\_MONTH\_NOTSTANDARD (v\_month)

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/126>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if dwc:month is either not present or is EMPTY; COMPLIANT if the value of dwc:month is an integer between 1 and 12

inclusive; otherwise NOT\_COMPLIANT

**Notes**: None

**INTERNAL\_PREREQUISITES\_NOT\_MET**: 370,003,397 (30.5%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_month IS NULL

**NOT\_COMPLIANT**: 2,039,919 (0.2%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_month IS NOT NULL

    AND (SAFE\_CAST(v\_month AS INT64)>12

    OR SAFE\_CAST(v\_month AS INT64)<1

    OR SAFE\_CAST(v\_month AS INT64) IS NULL)

**COMPLIANT**: 841,366,679 (69.3%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE SAFE\_CAST(v\_month AS INT64)<=12

    AND SAFE\_CAST(v\_month AS INT64)>=1

Test: TG2\_VALIDATION\_MONTH\_NOTSTANDARD (month)

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/126>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if dwc:month is either not present or is EMPTY; COMPLIANT if the value of dwc:month is an integer between 1 and 12

inclusive; otherwise NOT\_COMPLIANT

**Notes**: None

**INTERNAL\_PREREQUISITES\_NOT\_MET**: 119,902,189 (9.9%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE month IS NULL

**NOT\_COMPLIANT**: 0 (0%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE month NOT BETWEEN 1 AND 12

**COMPLIANT**: 1,093,507,806 (90.1%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE month BETWEEN 1 AND 12

Test: TG2\_VALIDATION\_DAY\_NOTSTANDARD (v\_day)

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/147>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if a) dwc:day is not present or is EMPTY, or b) dwc:day can not be cast as an integer, or c) dwc:day can be cast as an integer between 29 and 31 inclusive and dwc:month can not be cast as an integer between 1 and 12, or d) dwc:month can be cast as the integer 2 and dwc:month can be cast as the integer 29 and dwc:year can not be cast as a valid ISO 8601 year; COMPLIANT e) if the value of the field dwc:day can be cast as an integer between 1 and 28 inclusive, or f) dwc:day can be cast as an integer between 29 and 30 and dwc:month can be cast as one of (4,6,9,11), or g) dwc:day can be cast as an integer between 29 and 31 and dwc:month can be cast as one of (1,3,5,7,8,10,12), or h) dwc:day can be cast as the integer 29 and dwc:month can be cast as the integer 2 and  dwc:year is a valid leap year (evenly divisible by 400 or (evenly divisible by 4 but not evenly divisible by 100)); otherwise NOT\_COMPLIANT

**Notes**: This test must take into account the given month and year, if present, to account

for leap years. This is part of a group of similar tests (#125, #130, #131).

**INTERNAL\_PREREQUISITES\_NOT\_MET**: 385,161,318 (31.7%)

    a) dwc:day is not present or is EMPTY 384,839,246 (31.7%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_day IS NULL

    b) dwc:day can not be cast as an integer 246,932 (0.0%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE v\_day IS NOT NULL

    AND safe\_cast(v\_day AS INT64) IS NULL

    c) dwc:day is between 29 and 31, but dwc:month is not between 1 and 12 75,128 (0.0%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE

    safe\_cast(v\_day AS INT64) BETWEEN 29 AND 31

    AND (safe\_cast(v\_month AS INT64) IS NULL

    OR safe\_cast(v\_month AS INT64) NOT BETWEEN 1 AND 12)

    d) dwc:day can be cast as the integer 29 and dwc:month can be cast as the integer 2, but dwc:year can not be cast as a valid ISO 8601 year 12 (0.0%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE safe\_cast(v\_month AS INT64)=2

    AND safe\_cast(v\_day AS INT64)=29

    AND (safe\_cast(v\_year AS INT64) IS NULL OR

    safe\_cast(v\_year AS INT64)>=2019)

**NOT\_COMPLIANT**: 1,753,274 (0.14%)

**COMPLIANT**: 826,495,403 (68.1%)

    e) dwc:day can be cast as an integer between 1 and 28 inclusive 760,216,494 (62.7%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE safe\_cast(v\_day AS INT64) BETWEEN 1 AND 28

    f) dwc:day can be cast as an integer between 29 and 30 and dwc:month can be cast as one of (4,6,9,11) 18,721,309 (1.5%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE safe\_cast(v\_day AS INT64) BETWEEN 29 AND 30

    AND safe\_cast(v\_month AS INT64) IN (4,6,9,11)

    g) dwc:day can be cast as an integer between 29 and 31 and dwc:month can be cast as one of (1,3,5,7,8,10,12) 47,173,157 (3.9%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE safe\_cast(v\_day AS INT64) BETWEEN 29 AND 31

    AND safe\_cast(v\_month AS INT64) IN (1,3,5,7,8,10,12)

    h) dwc:day can be cast as the integer 29 and dwc:month can be cast as the integer 2 and  dwc:year is a valid leap year (evenly divisible by 400 or (evenly divisible by 4 but not evenly divisible by 100)) 384,443 (0.0%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE safe\_cast(v\_day AS INT64)=29

    AND safe\_cast(v\_month AS INT64)=2

    AND safe\_cast(v\_year AS INT64)<=2019

    AND

    ( MOD(safe\_cast(v\_year AS INT64),400)=0

      OR

      ( MOD(safe\_cast(v\_year AS INT64),4)=0

        AND MOD(safe\_cast(v\_year AS INT64),100)<>0

      )

    )

Test: TG2\_VALIDATION\_DAY\_NOTSTANDARD (day)

**GitHub Issue:** <https://github.com/tdwg/bdq/issues/147>

**Expected response**: INTERNAL\_PREREQUISITES\_NOT\_MET if a) dwc:day is not present or is EMPTY, or b) dwc:day can not be cast as an integer, or c) dwc:day can be cast as an integer between 29 and 31 inclusive and dwc:month can not be cast as an integer between 1 and 12, or d) dwc:month can be cast as the integer 2 and dwc:month can be cast as the integer 29 and dwc:year can not be cast as a valid ISO 8601 year; COMPLIANT e) if the value of the field dwc:day can be cast as an integer between 1 and 28 inclusive, or f) dwc:day can be cast as an integer between 29 and 30 and dwc:month can be cast as one of (4,6,9,11), or g) dwc:day can be cast as an integer between 29 and 31 and dwc:month can be cast as one of (1,3,5,7,8,10,12), or h) dwc:day can be cast as the integer 29 and dwc:month can be cast as the integer 2 and  dwc:year is a valid leap year (evenly divisible by 400 or (evenly divisible by 4 but not evenly divisible by 100)); otherwise NOT\_COMPLIANT

**Notes**: This test must take into account the given month and year, if present, to account

for leap years. This is part of a group of similar tests (#125, #130, #131).

**INTERNAL\_PREREQUISITES\_NOT\_MET**: 131,540,410 (10.8%)

    a) dwc:day is not present or is EMPTY 131,540,410 (10.8%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE day IS NULL

    b) dwc:day can not be cast as an integer 0 (0%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE day IS NOT NULL

    AND safe\_cast(day AS INT64) IS NULL

    c) dwc:day is between 29 and 31, but dwc:month is not between 1 and 12 0 (0%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE day BETWEEN 29 AND 31

    AND (month IS NULL

    OR month NOT BETWEEN 1 AND 12)

    d) dwc:day can be cast as the integer 29 and dwc:month can be cast as the integer 2, but dwc:year can not be cast as a valid ISO 8601 year 0 (0.0%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE month=2

    AND day=29

    AND (year IS NULL OR year>=2019)

**NOT\_COMPLIANT**: 0 (0%)

**COMPLIANT**:  1,081,869,585 (89.2%)

    e) dwc:day can be cast as an integer between 1 and 28 inclusive 998,258,501 (82.3%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE day BETWEEN 1 AND 28

    f) dwc:day can be cast as an integer between 29 and 30 and dwc:month can be cast as one of (4,6,9,11) 23,854,989 (2.0%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE day BETWEEN 29 AND 30

    AND month IN (4,6,9,11)

    g) dwc:day can be cast as an integer between 29 and 31 and dwc:month can be cast as one of (1,3,5,7,8,10,12) 59,265,535 (4.9%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE day BETWEEN 29 AND 31

    AND month IN (1,3,5,7,8,10,12)

    h) dwc:day can be cast as the integer 29 and dwc:month can be cast as the integer 2 and  dwc:year is a valid leap year (evenly divisible by 400 or (evenly divisible by 4 but not evenly divisible by 100)) 490,560 (0.0%)

    SELECT sum(occcount) AS occcounts FROM `gbif\_dates`

    WHERE day=29

    AND month=2

    AND year<=2019

    AND

    ( mod(year,400)=0

      OR

      ( mod(year,4)=0

        AND mod(year,100)<>0

      )

    )