# Datetime handling

## Summary

Guidance and examples for using date and time values in instructions.

**On this page:**

## Datetime Functions

In addition to those functions provided by SQLite, the Agent language has the following date functions for use in SELECT statements:

- DATETRUNC — Truncates a date-time value, expressed as a Unix epoch time integer, to a given resolution and returns a corresponding Unix epoch time integer.
- EPOCHTOJSON — Takes a date-time value, expressed as a Unix epoch time in UTC, and returns a JSON-compatible (ISO-8601) string representation of that date-time, also in UTC.

## Formats

Irrespective of the local timezone or daylight savings adjustment, the Agent always presents datetime as UTC unless a function is used to specifically convert to local time.

If you want to know the device's timezone, use the Agent.GetSummary method, which reports current TimeZone as an offset from UTC in minutes (positive or negative). If using TIMS, then you will need to restart TIMS to read any changes in timezone.

Agent methods return datetime columns as either

- UnixEpoch, defined as the number of seconds that have elapsed since 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970, which may be either SQL datatype integer or string
- IS0-8601 string YYYY-MM-DDTHH:MM:SSZ in which the timezone offset is always Z because Agent time is always presented in UTC.

### Converting from Unix Epoch timestamp into readable formats

As previously stated, SQLite has many date and time functions, documented online at: https://sqlite.org/lang_datefunc.html. The Agent language also provides the EPOCHTOJSON function to convert a UnixEpoch integer to a IS0-8601 string. The example below shows how the SQLite functions STRFTIME and DATETIME could be used instead. It also shows these functions do not require the TS string to be CAST as an integer. Finally, the example also shows how DATETIME uses TimeZone.

**Example used to display Unix Epoch timestamp in different formats**

```
// Get TimeZone and current time
@summary = Agent.GetSummary();
@now = SELECT STRFTIME("%s","now") AS TS;
// Next statement shows that TS string can be CAST as an integer
@now = SELECT TS, CAST(TS AS INTEGER) AS TS_int, TimeZone FROM @now, @summary;
// Now show datatypes and conversions
SELECT
  TS,
  EPOCHTOJSON(TS) AS TS_EpochToJson,
  DATETIME(TS, "unixepoch") as TS_datetime,
  DATETIME(TS, "unixepoch", "localtime") as TS_localtime,
  STRFTIME("%Y-%m-%dT%H:%M:%SZ",DATETIME(TS, "unixepoch")) as TS_strftime,
  TYPEOF(TS),
  TYPEOF(TS_int),
  TimeZone
FROM @now;
```

The above example instruction will return something similar to the following in TIMS.

| TS | TS_EpochToJson | TS_datetime | TS_localtime | TS_strftime | TYPEOF(TS) | TYPEOF(TS_int) | TimeZone |
|---|---|---|---|---|---|---|---|
| 1515972745 | 2018-01-14T23:32:25Z | 2018-01-14 23:32:25 | 2018-01-14 18:32:25 | 2018-01-14T23:32:25Z | text | integer | -300 |

### Converting from readable format into Unix Epoch timestamp

**Example used to convert ISO-8601 formatted string into Unix epoch time**

```
// Converts datetime string "2017-08-18T19:18:03Z" or "2017-08-18 19:18:03" into unixepoch integer
1503083883.
// STRFTIME returns the integer as a TEXT datatype; if we want NUMERIC datatype then we need to CAST
@unixepoch = SELECT CAST(STRFTIME("%s","2017-08-18T19:18:03Z") AS INTEGER) AS TS;
```

## Making data lists

**Example to create list of last 30 dates**

```
@now = SELECT STRFTIME("%s","now") AS TS;
@seq = Utilities.GenerateSequence(Limit: 30);
FOREACH @i IN @seq
DO
    @then = SELECT @now.TS-((@i.Sequence-1)*86400) AS TS FROM @now, @i;
    SELECT STRFTIME("%Y-%m-%d",DATETIME(TS, "unixepoch")) AS Date FROM @then;
DONE;
```

**Example to create list between two dates**

```
@dates = SELECT "2018-01-10" AS D1, "2018-01-14" AS D2;
@days = SELECT CAST(JULIANDAY(D2) - JULIANDAY(D1) + 1 AS INTEGER) AS Diff FROM @dates;
@seq = Utilities.GenerateSequence(Limit: @days.Diff);
FOREACH @i IN @seq
DO
    SELECT STRFTIME("%Y-%m-%d", (JULIANDAY(D1) + @i.Sequence-1)) AS Date FROM @dates, @i;
DONE;
```

## When is now?

The examples above show how STRFTIME can tell you when an instruction was run, as UTC and adjusted to local time using TimeZone.

Sometimes it is useful for an instruction to know when it was originally sent, which may have been many moments before it was run. SCALE provides this with the following Environment variable:

| | |
|---|---|
| %environment:now% | UTC ISO-8601 compatible date time stamp of when the instruction was issued. ⚠ This is the timestamp of when the instruction was originally run and sent by the Tachyon Server. It remains the same if the same instruction is re-run. |

**Example to show difference of when an instruction is run and when it was sent**

```
// these times are the same in TIMS
SELECT EPOCHTOJSON(STRFTIME("%s","now")) AS TS_run, "%environment:now%" AS TS_sent;
```