



LANDING.AERO / RESEARCH & PRACTICAL GUIDANCE

SSIM Schedule Times to UTC: DST Acceptance Test Matrix

LANDING.AERO is a Montreal software studio that builds custom, AI-assisted software for flight operations: operations tools, data integration, workflow automation and dashboards designed around how an airline, charter operator or flight department actually works.

Published by LANDING.AERO · 2026-10-19 · ssim schedule times to utc · ssim utc conversion · daylight saving time · airline schedule integration · iana tzdb · flight operations · utc acceptance tests · schedule data quality

Original article: <https://www.landing.aero/articles/en/ssim-schedule-times-utc-dst-test-matrix>



In this article

1. Executive Summary
 2. Introduction and Background
 3. Key Changes
 4. Implementation Considerations and Process Changes
 5. The UTC Acceptance Test Matrix
 6. Evaluating a Packaged Importer or a Narrow Bridge
 7. Data Analysis and Evidence
 8. Implications and Future Directions
 9. Conclusion
-

Executive Summary

To convert **Standard Schedules Information Manual (SSIM) schedule times to Coordinated Universal Time (UTC)**, first establish the operating date, station, declared time mode and applicable source profile. For a local-time event, resolve the dated station time against a governed time-zone mapping and a recorded rules version. Only then apply **UTC instant = resolved local date-time minus its date-specific offset**. A local clock reading can identify no instant or multiple instants around a clock change; subtraction alone cannot settle those cases. The Internet Engineering Task Force defines the offset arithmetic and documents this ambiguity explicitly. (www.rfc-editor.org)

The current standards context makes version control material. The International Air Transport Association's **36th-edition, 2026 SSIM** public page lists Appendix F UTC/local comparisons and daylight saving time (DST) information for **2026-2027 and 2028**. The Internet Assigned Numbers Authority (IANA) lists **time-zone database (tzdb) 2026e**, released **September 29, 2026**, including a forthcoming Manitoba change. These publications have different roles: the licensed SSIM defines the exchange contract, while the time-zone database supplies implementation rules for named zones. Neither a country code nor an airport code alone establishes the conversion contract. (www.iata.org) (www.iana.org) (openflights.org)

This report provides **10 synthetic acceptance cases**, covering ordinary UTC and local input, a spring gap, an unresolved autumn overlap, explicitly disambiguated overlaps, date rollover, different station transition dates, a fractional offset and a half-hour transition. The calculations were executed for this report against the container's measured **2026b** system database, with **12 event assertions passing**. That measured baseline must be distinguished from current-release conformance. Python can use the operating system's database, so the application version alone does not identify the rules actually executed. (docs.python.org)

The recommended interface policy is to **reject nonexistent local times, hold unresolved ambiguous times for review**, and publish only resolved events with their original input and provenance. These are proposed acceptance requirements, not quotations from licensed SSIM. Java and PostgreSQL document automatic adjustment or ambiguity resolution, making successful parsing an insufficient acceptance result. For downstream context, the United States Federal Aviation Administration (FAA) requires UTC departure time in its international flight-plan instructions; that requirement does not establish crew or dispatch legality. (docs.oracle.com) (www.postgresql.org) (www.faa.gov)

Introduction and Background

The difficult part of schedule conversion is establishing what the input means. A dated UTC event and a recurring station-local departure can look similar in a display while requiring different processing. An importer should preserve that distinction before it creates operational events. SSIM, the **Standard Schedules Information Manual**, is IATA's schedule-exchange standard; its public contents include schedule data requirements, schedule-set transfer and UTC/local comparisons. This report addresses the boundary between that exchange and an operator's dated operational records. (www.iata.org)

The intended readers are schedule-data owners, **operations control centre (OCC)** managers, crew-system integrators and the technical teams supporting dispatch interfaces. Their acceptance question is concrete: can the interface produce an unambiguous UTC instant, explain how it obtained that instant, and stop when the supplied evidence is insufficient? A standards-based reason for asking is that local-to-instant conversion is not always a function with a single result. Offset transitions can create missing or repeated local readings. (www.rfc-editor.org)

The examples below are explicitly **synthetic acceptance fixtures**, not published airline services. Airport labels, where used, are fixture identifiers with declared zone mappings. They do not assert that a public airport database has been approved for an operator's operational use. OpenFlights documents separate airport-code, numeric-offset and named-zone fields; GeoNames documents a coordinate-based zone lookup. Both illustrate possible mapping inputs, rather than a universal rule that converts a three-letter airport code into an instant. (openflights.org) (www.geonames.org)

LANDING.AERO's position in this decision is that of a custom integration builder. It is considered only in the commissioned-bridge comparison, where a buyer would specify the same acceptance evidence required from a packaged importer. The report does not rank vendors by undocumented capabilities. Its output is a data-quality specification: a test corpus, explicit error states, and a handoff record that the operator can assess within its existing procedures.

Key Changes

Separate the licensed exchange standard from the implementation profile

IATA's public **2026** changes list explicitly updates Appendix F daylight saving information for **2026-2027 and 2028**. That establishes the relevance of current civil-time information to schedule exchange. It does not publicly establish every record position, offset sign convention, default, arrival-day relationship or override precedence needed to implement a particular feed. Those details should be checked against the licensed edition and the sender's interface agreement before acceptance. No licensed SSIM text is reproduced here. (www.iata.org)

A useful public implementation example is Avinor's mapping, last updated **May 10, 2021**, which identifies **U as UTC and L as local time**. It supports the need to carry a declared mode through the pipeline. It is an implementation-specific, dated mapping, not a substitute for current licensed field definitions. A source profile should therefore identify the exact message or record family and explain how its dates, time mode and offsets become the normalized inputs used below. (xwiki.avinor.no)

The practical change is contractual: do not let a parser's successful field extraction stand in for an agreed semantic interpretation. A test should identify the original record, the profile version and the normalized event. When the profile is missing, the appropriate result is a hold for interpretation, even if the clock digits look valid.

Treat time-zone rules as versioned reference data

As accessed on **October 19, 2026**, IANA lists **2026e**, released **September 29, 2026**. Its release notes identify Manitoba moving to permanent **UTC-05:00 on October 31, 2026**. This is a concrete reason to distinguish a previously generated future instant from one recomputed under updated rules. Recording a zone name without the rules version leaves an important part of the calculation unspecified. (www.iana.org)

The Canadian National Research Council's page also warns that some information will cease to be accurate from **November 1, 2026**. That warning should not be generalized to every Canadian station or used to infer a replacement rule. It demonstrates why a current-looking reference page and a pinned machine-readable rules release must be evaluated separately. IANA notes that governments can change time rules with little notice. (nrc.canada.ca) (www.iana.org)

For acceptance, identify the deployed rules artifact rather than recording only the programming-language version. Python's documentation says its zone-information module uses system time-zone data when available. A deployment that updates the runtime but leaves the operating-system data unchanged can therefore have different provenance from one that updates both. (docs.python.org)

Make ambiguity an explicit outcome

The model should allow **zero, one or multiple candidate instants** before it selects an output. Java's zone-rules application programming interface (API) explicitly distinguishes normal times, gaps and overlaps through the number of valid offsets. Microsoft's APIs separately identify ambiguous and invalid times. These facilities support detection, but the interface must still define what happens next. (docs.oracle.com) (learn.microsoft.com)

The proposed policy is conservative and reviewable: a gap produces no published instant; an unresolved overlap retains its candidates and waits for authoritative disambiguation. An explicit offset can settle an overlap only if it is valid for that zone and local date-time under the selected rules. A supplied offset that disagrees with the zone should trigger an explicit conflict state rather than silent precedence. (www.rfc-editor.org)

That policy belongs to this acceptance specification. Request for Comments (RFC) 9557 discusses local-time ambiguity and offset/zone inconsistencies in its own timestamp format; its requirements should not be represented as SSIM requirements. Likewise, a library default that chooses an overlap occurrence is a documented behavior to test, not evidence that the sender intended that occurrence. (www.rfc-editor.org) (docs.oracle.com)

Implementation Considerations and Process Changes

Establish a complete event before conversion

Table 1 defines the proposed normalized input contract. These are interface fields, not SSIM field names or byte positions. The source-profile mapping must establish how the licensed source supplies each value. The distinction matters because Avinor's mapping separates schedule validity, creation dates and operation dates, illustrating that a date elsewhere in the record is not automatically the event's operating date. (xwiki.avinor.no)

Input	Required interpretation	Acceptance evidence
Operating date and local clock	Full calendar date for this event at this station	Retain the source representation and date derivation
Declared time mode	UTC or station-local under the agreed source profile	Reject unknown tokens; hold missing declarations
Station identity	Airport or station in the sender's namespace	Versioned mapping, including effective dates
Named zone	Governed IANA identifier when civil rules are needed	Mapping source, override and approval history
Explicit offset or occurrence	Optional disambiguator with defined precedence	Must match a valid candidate when a zone is also asserted
Arrival-day relationship	Sender-defined relationship used to construct arrival date	Never infer it solely from clock ordering
Source profile and edition	Record-family semantics and agreed version	Licensed specification reference and profile identifier
Rules and converter versions	Actual data and code artifacts used	Deployment identifier and reproducible execution record

The table makes missing context visible before arithmetic starts. The source record should remain available beside the normalized fields, so a corrected profile or mapping can be replayed without guessing the original input. A display name should also remain separate from the zone identifier: Unicode distinguishes stable identifiers from localized names and documents canonical-name stability differences between its data and the underlying database. (unicode.org)

The intake procedure should require the following:

- **Expand dates:** Generate each operating event from its validity and day-of-operation rules before resolving station-local time.
- **Read mode:** Interpret the declared mode using the agreed source profile, preserving the original token.
- **Map station:** Resolve the station through the effective mapping snapshot and record any override.
- **Resolve arrival:** Construct the arrival calendar date from the source's agreed day relationship.
- **Check candidates:** Enumerate valid offsets for the complete local event before selecting an instant.
- **Record provenance:** Attach source revision, mapping version, rules version and converter build to the result.

These steps are proposed workflow requirements. The technical reason for separating candidate detection from conversion is documented across libraries: Java exposes valid-offset enumeration, while Microsoft exposes distinct invalid-time and ambiguous-time checks. Their documentation should guide the implementation appropriate to the selected runtime and input type. (docs.oracle.com) (learn.microsoft.com)

Define offset interpretation and conflict precedence

For a resolved local event, the sign is straightforward: **UTC = local minus offset**. Thus a negative offset moves the UTC result later, while a positive offset moves it earlier, potentially onto another calendar date. This arithmetic follows RFC 3339's definition of numeric offsets. It applies only after the input's local meaning and valid offset have been established. (www.rfc-editor.org)

Do not assume that a field labeled "UTC variation" in an arbitrary feed uses the same encoding as an RFC timestamp. The interface profile should state its units, sign, missing-value representation and applicability to departure or arrival. If a feed carries both a civil zone and an offset, document whether the offset is a required assertion or whether the profile explicitly authorizes it to override civil rules. A conflict without such a contract should be held.

The offset should not be used to recover an original station zone. Many zones can share an offset at a particular instant, while their future rules differ. GeoNames itself distinguishes the named zone from its standard-time raw offset; its standard offset is not a date-specific DST answer. (www.geonames.org)

Give exceptions durable states

The proposed interface should expose at least these states:

- **ACCEPT:** Exactly one candidate remains after valid disambiguation and all input checks.
- **REJECT_INVALID_LOCAL:** No instant corresponds to the supplied local date-time under the asserted rules.
- **HOLD_AMBIGUOUS:** Multiple candidates remain and the source has not identified the intended occurrence.
- **HOLD_CONTEXT:** Date, time mode, station mapping or profile meaning is missing.
- **HOLD_CONFLICT:** Supplied date, offset, zone or occurrence assertions disagree.
- **SUPERSEDED:** A previously accepted result has been replaced by a traceable source or reference-data revision.

These labels are proposed interface states, not official SSIM codes. Their value is operational visibility: a rejected event should not quietly enter a downstream queue, while a held event should retain enough information for the schedule-data owner to resolve it. PostgreSQL documents that syntactically valid gap and overlap inputs can be accepted through automatic interpretation, so a database insert alone cannot certify this policy. (www.postgresql.org)

The UTC Acceptance Test Matrix

Fixture scope and provenance

All rows in *Table 2* are **synthetic examples**, expressed in normalized interface notation. They are not raw SSIM records. Each assumes an agreed profile, complete event dates and declared fixture-zone mappings. The test run used Python's zone-information interface and the container's system database, whose local version header read **2026b**. Those are measured execution facts from this report, not claims that the vendor or current IANA release uses that baseline. Python documents the dependency on system data. (docs.python.org)

For civil-rule cross-checks, NIST lists United States DST changes on **March 8 and November 1, 2026**; the UK lists **March 29 and October 25, 2026**. These different dates justify a fixture between the respective spring changes. The New South Wales (NSW) government lists its spring change on **October 4, 2026**, and separately documents Lord Howe Island's **30-minute** adjustment. The dated Lord Howe gap below is a pinned-database calculation, rather than an extrapolation of mainland clock wording. (www.nist.gov) (www.gov.uk) (www.nsw.gov.au) (dcj.nsw.gov.au)

Case and input	Zone/source basis	Expected UTC result in measured fixture	Required state and evidence
T1: UTC mode, 2026-01-15 10:00	Declared UTC; no local conversion	2026-01-15 10:00Z	ACCEPT; retain UTC declaration (www.rfc-editor.org)
T2: Local mode, 2026-01-15 10:00	America/New_York; system 2026b	2026-01-15 15:00Z	ACCEPT; resolved offset -05:00 (www.nist.gov)
T3: Local mode, 2026-03-08 02:30	America/New_York; spring gap	No candidate	REJECT_INVALID_LOCAL; retain original value (www.nist.gov)
T4: Local mode, 2026-11-01 01:30; no occurrence	America/New_York; autumn overlap	Candidates 05:30Z and 06:30Z on 2026-11-01	HOLD_AMBIGUOUS; publish neither (www.nist.gov)
T5: T4 with explicit -04:00	Same zone; offset must be valid	2026-11-01 05:30Z	ACCEPT; retain disambiguating source assertion (www.nist.gov)
T6: T4 with explicit -05:00	Same zone; offset must be valid	2026-11-01 06:30Z	ACCEPT; retain disambiguating source assertion (www.nist.gov)
T7: Departure 2026-03-21 23:30 New York; arrival 2026-03-22 11:30 London	America/New_York and Europe/London; explicit arrival date	Departure 2026-03-22 03:30Z; arrival 2026-03-22 11:30Z	ACCEPT; departure UTC date rolls forward (www.nist.gov) (www.gov.uk)
T8: Independent station events, each 2026-03-22 09:00 local	America/New_York and Europe/London; different DST calendars	New York 2026-03-22 13:00Z; London 2026-03-22 09:00Z	ACCEPT; resolve each station separately (www.nist.gov) (www.gov.uk)
T9: Local mode, 2026-06-15 10:00	Pacific/Chatham; standard offset +12:45	2026-06-14 21:15Z	ACCEPT; fractional offset and previous UTC date (www.govt.nz)
T10: Local mode, 2026-10-04 02:15	Australia/Lord_Howe; pinned 2026b half-hour gap	No candidate	REJECT_INVALID_LOCAL; do not round or shift (dcj.nsw.gov.au) (www.nsw.gov.au)

The expected values are this report's derived calculations, not times published by an airline or outputs measured from a commercial importer. The Chatham fractional-offset premise has an independent government basis: New Zealand standard time is **UTC+12**, and the Chatham Islands are **45 minutes** ahead. The table combines that premise with its explicitly dated, pinned-rule fixture. (www.govt.nz)

Reproduce the candidate test

The executable method is to construct the naive local date-time and load its named zone. Try the runtime's possible overlap interpretations, convert each candidate to UTC, convert back to the same zone, and retain only candidates whose returned local date and clock exactly match the input. Deduplicate the remaining instants. If an explicit offset is supplied, retain only candidates with that offset. In Python, the overlap interpretations use **fold=0** and **fold=1**, whose documented offsets differ across an overlap. (docs.python.org)

This round-trip check matters for a gap because merely attaching a zone need not prove that the local time existed. The acceptance assertion should compare candidate sets before checking the final business state. Java's documented zero/one/two valid-offset model offers another implementation route; Microsoft's invalid and ambiguous checks provide separate detection mechanisms. A team should implement the equivalent procedure for its runtime and confirm its edge semantics. (docs.oracle.com) (learn.microsoft.com)

The executed run asserted **12 event results across the 10 rows**, because the cross-midnight and mixed-station rows each contain separate events. All assertions passed against the stated baseline. An unresolved overlap passes its test by returning the expected candidate set and holding publication; a gap passes by returning no candidate and rejecting publication. Producing a timestamp for every row would fail this specification.

Extend the corpus beyond the core fixtures

The compact matrix should be augmented with the operator's actual station mappings, source profiles and season boundaries. Keep the synthetic fixtures stable so regressions remain comparable, then add dated real-format records under the applicable licensing arrangements. A real-format record should establish that the parser derives the same normalized inputs, not simply that a downstream date library can convert a hand-entered timestamp.

Additional tests should cover:

- **Missing mode:** A plausible clock value with no agreed interpretation must remain held.
- **Unknown station:** An unmapped code must not fall back to the server's local zone.
- **Offset disagreement:** An asserted offset outside the valid candidate set must remain held.
- **Absent arrival date:** Clock ordering alone must not manufacture the arrival-day relationship.
- **Duplicate import:** Reprocessing the same revision must preserve event identity and provenance.
- **Rules update:** Recalculation must expose changed future instants before downstream replacement.

These extensions test interface behavior that a conceptual local/UTC explanation cannot establish. Active Flights' article claims its toolkit honors declared mode and date-specific offsets. That is relevant vendor documentation, but the opened article does not independently demonstrate the strict gap, overlap or conflict states proposed here. Those outcomes should be established by executing the corpus. (activeflights.com)

Evaluating a Packaged Importer or a Narrow Bridge

Compare evidence at the same boundary

A packaged product, an incumbent operations system and a commissioned adapter should receive the same normalized cases and representative licensed-format records. Evaluate the returned instant, state and provenance together. A product that selects a documented default may be suitable if the operator's source contract explicitly supplies that policy; it should not be credited with resolving an intention that the source never expressed.

Table 3 compares procurement paths. LANDING.AERO is included as a **custom bridge option**, consistent with its role in integration work. The row describes an evaluation scope to commission, not a claim that an existing product has already passed these tests. Active Flights is the named packaged-tool example because its own article describes relevant behavior. (activeflights.com)

Option	Scope to evaluate	Evidence required before acceptance
Active Flights SSIM Toolkit	Documented declared-mode and date-specific offset handling	Execute the corpus; obtain actual gap, overlap, conflict and version outputs
Existing schedule or operations vendor	The deployed importer, configuration and supported source profiles	Version-specific results plus documented exception and replay behavior
LANDING.AERO, commissioned custom bridge	A specified schedule-to-operations interface around the operator's contract	Deliverable tests, controlled mappings, exception workflow and deployment provenance

The comparison should not become a feature-ranking exercise based on missing public documentation. For the named toolkit, public statements establish a relevant capability claim, while tested outputs establish conformity to the buyer's policy. For a commissioned bridge, the acceptance corpus belongs in the delivery specification. For the incumbent path, run the deployed configuration rather than assuming the current documentation describes a historical installation.

Ask questions that produce inspectable answers

A vendor evaluation should request concrete artifacts:

- **Time interpretation:** Show how each supported source profile establishes mode and event date.
- **Ambiguity policy:** Return all candidates before applying a documented occurrence-selection rule.
- **Reference data:** Identify the deployed zone database and effective station-mapping snapshot.
- **Error transport:** Show how held and rejected records appear in APIs, queues and operator screens.
- **Replay controls:** Demonstrate how a corrected input or new rules artifact supersedes earlier output.
- **Downstream contract:** Show the complete event payload, including original local values and provenance.

These requests focus on observable behavior. Java's convenience conversion can choose the earlier overlap offset and shift a gap forward; PostgreSQL also documents automatic gap and overlap interpretation. Neither behavior is an adverse product finding. They are neutral defaults that an acceptance layer must account for

when enforcing a stricter contract. (docs.oracle.com) (www.postgresql.org)

A narrow bridge is justified when the needed scope is normalization, reference-data control, exception handling and handoff to established systems. It should not be treated as a replacement for a crew-scheduling, dispatch or flight-planning product when those are the capabilities the operator actually needs. The acceptance boundary should state where schedule normalization ends and each downstream system's own responsibilities begin.

Data Analysis and Evidence

What the measured results establish

The quantitative core is a deterministic fixture exercise, not a survey, market estimate or vendor benchmark. The measured **10-row corpus** contains **12 event assertions**: **8 rows** contain a single event, and **2 rows** contain paired events. The expected business outcomes are **7 accepted rows, 2 invalid-local rejections and 1 ambiguous hold**. These counts are derived directly from Table 2 and the executed assertions.

That composition tests more than ordinary offset subtraction. The unresolved autumn case has **2 candidates separated by 60 minutes**; the explicitly disambiguated cases select each candidate in turn. The Chatham example uses a **45-minute fractional component**, while Lord Howe's published transition size is **30 minutes**. The arithmetic and candidate counts are report calculations; their rule premises come from the pinned database and the relevant primary documentation. (www.nist.gov) (www.govt.nz) (dcj.nsw.gov.au)

The cross-midnight fixture also verifies complete dates rather than clock-only values. Its elapsed UTC interval is **8 hours**, calculated from **03:30Z to 11:30Z on March 22**. This is a consistency check for that synthetic leg, not a claim about a scheduled route or acceptable flight duration. The arithmetic uses fully resolved instants, following the offset convention defined in RFC 3339. (www.rfc-editor.org)

Why station-specific calendars belong in the test set

The United States spring transition on **March 8, 2026** and UK transition on **March 29, 2026** are **21 days** apart. The mixed-station fixture falls inside that interval. It demonstrates why applying a departure station's seasonal assumption to an arrival station is insufficient. The dates come from national primary sources; the interval is a calendar subtraction performed for this report. (www.nist.gov) (www.gov.uk)

Southern-hemisphere dates add another independent check. New Zealand lists its DST start on **September 27, 2026**, while NSW lists **October 4, 2026**, a **7-day** difference. A test strategy based solely on a common "summer schedule" label would miss those distinct civil calendars. The label attached to an airline schedule season should not replace a dated rule lookup at each station. (www.govt.nz) (www.nsw.gov.au)

Airport reference data introduces a separate freshness issue. OpenFlights says its DST field is a country-level approximation for **2009** and describes its repository copy as sporadically updated. Its named-zone field can be useful mapping evidence, but the documented DST approximation should not be promoted into current date-specific rules. GeoNames similarly exposes a standard offset separately from the rule identifier. (openflights.org) (www.geonames.org)

Boundaries of the evidence

The public evidence establishes the standards context, civil-rule dates and documented library behaviors. It does not independently establish a commercial importer's results on this corpus, prove that a public mapping is operationally approved, or verify licensed record-field semantics. Those are explicit acceptance tasks for the deployment under review.

The report's **2026b** run also does not certify **2026e** conformance. Current-release testing requires execution against the selected production artifact, followed by comparison with the retained baseline. IANA's release information establishes what is current; Python's documentation explains why the local execution environment must be inspected separately. Reporting both facts preserves the distinction between source currency and measured implementation behavior. (www.iana.org) (docs.python.org)

Implications and Future Directions

Preserve both the instant and its explanation

A downstream event should carry the accepted UTC instant together with the source-local date and clock, station, declared mode, resolved offset, source revision, mapping version, rules version and ambiguity decision. This is a proposed handoff schema. It makes it possible to explain a changed instant without reconstructing what the sender originally supplied.

Database design should support that retention explicitly. PostgreSQL documents that its timestamp-with-time-zone value does not retain the originally stated or assumed zone. Therefore, a timestamp column by itself cannot preserve the full conversion explanation. Separate provenance fields or an associated immutable source record are needed if that explanation is an acceptance requirement. (www.postgresql.org)

For the United States flight-plan boundary, FAA instructions specify UTC departure time in **Item 13** and identify a **DOF**/ date-of-flight field in the instructions and tables. The interface should hand over a complete dated event so a receiving flight-planning system can apply its own filing requirements. This report does not turn a schedule event directly into an approved flight plan or assess flight, duty or dispatch legality. (www.faa.gov)

Govern recalculation as a visible change

Future local schedules need a controlled response when civil rules change. The W3C time-zone document, explicitly a **Group Draft Note and work in progress**, discusses refreshing precomputed future values after runtime database updates. Its advice is useful implementation context, while IANA's release stream supplies concrete evidence that reference rules change. (www.w3.org) (www.iana.org)

The handoff workflow should require:

- **Retain input:** Keep original local and source representations alongside the accepted instant.
- **Attach revision:** Identify the source, mapping, rules and converter artifacts used.
- **Compare outputs:** Show changed future instants before replacing accepted records.
- **Control publication:** Prevent held or rejected events from appearing as resolved operational times.
- **Preserve lineage:** Link every replacement to the result it supersedes.

- **Reconcile consumers:** Confirm downstream systems receive the accepted revision and acknowledge changes.

Automation can execute these checks and present exceptions consistently. Selection of a missing operating date, an undocumented station mapping or an unresolved overlap remains a source-contract decision. The interface should ask for that decision through its exception workflow rather than manufacture it during conversion.

Conclusion

Correct SSIM-to-UTC conversion starts with a complete meaning for the source event: its operating date, station, declared time mode and source-profile semantics. The acceptance boundary should then resolve the event using governed mappings and recorded rules, retaining enough evidence to reproduce the result. An apparently reasonable clock value should not bypass missing context. (www.rfc-editor.org)

The proposed matrix gives buyers and integrators a concrete basis for that evaluation. It tests accepted instants alongside invalid and ambiguous outcomes, complete calendar dates alongside clock values, and station-specific rules alongside fractional offsets. Its expected results are derived synthetic fixtures with a stated execution baseline. They are neither a reproduction of licensed SSIM nor a certification of a vendor implementation.

Procurement should compare packaged importers, incumbent configurations and commissioned bridges using the same corpus and handoff requirements. Public documentation can establish what a vendor claims, but acceptance requires inspectable output from the actual deployed configuration. Missing evidence should become a test request, without being converted into an unsupported product judgment.

The durable outcome is a dated UTC event with its original local representation and conversion lineage. (www.postgresql.org) That record allows dispatch, crew and flight-planning interfaces to receive a resolved value while preserving the information needed to explain, correct and supersede it. The source owner remains able to distinguish a changed schedule from a changed civil-time rule, and downstream systems can apply their own procedures to the accepted revision.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. <https://www.rfc-editor.org/rfc/rfc3339.html>
2. <https://www.iata.org/en/publications/manuals/standard-schedules-information>
3. <https://www.iana.org/time-zones>
4. <https://openflights.org/data.php>
5. <https://docs.python.org/3/library/zoneinfo.html>
6. <https://docs.oracle.com/en/java/javase/25/docs/api/java.base/java/time/LocalDateTime.html>
7. <https://www.postgresql.org/docs/current/datetime-invalid-input.html>
8. https://www.faa.gov/air_traffic/publications/atpubs/fs_html/appendix_a.html

9. <https://www.rfc-editor.org/rfc/rfc9557.html>
10. <https://www.geonames.org/export/web-services.html>
11. <https://xwiki.avinor.no/spaces/ADDMIFTM/pages/45711533/SSIM%2B-%2BStandard%2BSchedule%2BInformation%2BMessage>
12. <https://nrc.canada.ca/en/certifications-evaluations-standards/canadas-official-time/time-zones-daylight-saving-time>
13. <https://www.iana.org/time-zones/tz-link>
14. <https://docs.oracle.com/en/java/javase/25/docs/api/java.base/java/time/zone/ZoneRules.html>
15. <https://learn.microsoft.com/en-us/dotnet/api/system.timezoneinfo.isambiguoustime?view=net-10.0>
16. <https://unicode.org/reports/tr35/tr35-dates.html>
17. <https://learn.microsoft.com/en-us/dotnet/api/system.timezoneinfo.isinvalidtime?view=net-10.0>
18. <https://www.nist.gov/pml/time-and-frequency-division/popular-links/daylight-saving-time-dst>
19. <https://www.gov.uk/when-do-the-clocks-change>
20. <https://www.nsw.gov.au/about-nsw/daylight-saving>
21. <https://dcj.nsw.gov.au/about-us/daylight-saving.html>
22. <https://www.nist.gov/pml/time-and-frequency-division/local-time-faqs>
23. <https://www.govt.nz/browse/recreation-and-the-environment/daylight-saving/governing-legislation>
24. <https://activeflights.com/blog/time-zones-utc-dst-schedules>
25. <https://www.govt.nz/browse/recreation-and-the-environment/daylight-saving>
26. <https://www.postgresql.org/docs/current/datatype-datetime.html>
27. <https://www.w3.org/TR/timezone>
28. <https://www.iana.org/time-zones/releases/2026e>

THE PUBLISHER

About LANDING.AERO

LANDING.AERO builds custom software for aviation operations teams at airlines, charter operators and flight departments. It is based in Montreal.

What it builds

Operations tools that reduce manual steps in day-to-day flight operations; integration of existing systems and data sources into one operational picture; workflow automation; real-time operational dashboards; bridges between legacy systems and modern applications; and mobile apps for staff in the field. Each application is designed around the operator's own workflows rather than sold as an off-the-shelf platform.

How it works

Work starts with discovery of the operator's workflows, continues with iterative development alongside the operator's team, and includes integration with the tools already in use and ongoing support.

Its own tools

LANDING.AERO also publishes small tools for aviation professionals, including UTC Alarm, which sets alarms in Zulu time, and NOTAM Notifier, announced for NOTAM change alerts on chosen airports.

Visit

See LANDING.AERO and its projects.

Articles published here explain general practice and cite their primary sources, naming the jurisdiction of any rule they describe. They are not operational, regulatory or legal advice, do not replace an operator's approved procedures or manuals, and requirements change.

Website: <https://www.landing.aero>

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.