



TECHNICAL HANDOFF

Booking: Cancellation fails with 'Credits over \$500 need a reason'

October 18, 2026 · 8 verified sources

The cancellation flow creates a credit with no Reason. A validation rule rejects any credit over \$500 without one, so large cancellations such as B-00000020 fail. The fix is to pass a reason into the credit.

Diagnosis

Root cause

Risk: The cancellation flow issues a credit through the Issue Resort Credit subflow, which creates a Credit record with an Amount and a Contact but never fills in Reason. The active validation rule Large_Credit_Needs_Reason rejects any Credit with Amount > 500 and a blank Reason, so every cancellation whose credit exceeds \$500 fails and the whole save is refused. Smaller cancellations pass because the rule only checks amounts above \$500. [1]

FAILING STEP

Create Credit Record

Issue_Resort_Credit, subflow of Booking: Cancellation

CREDIT FORMULA

Price × guests × 0.5

rounded to whole dollars

FAILS WHEN

Credit > \$500

booking value above about \$1,000

CONFIDENCE

High

from source; Culinary price not verified

How the failure happens, step by step

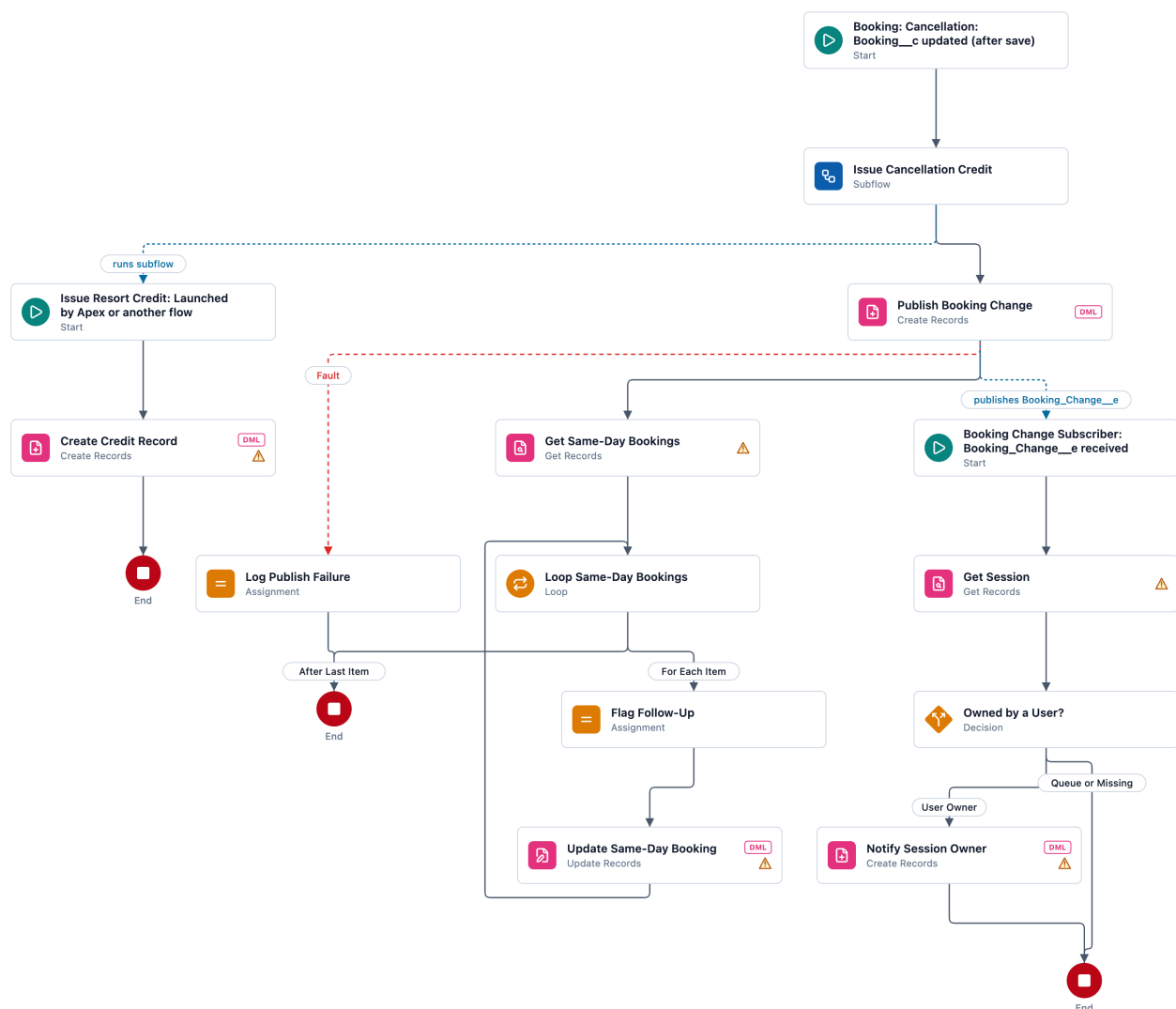
1. Front desk marks booking B-00000020 canceled. Booking: Cancellation is an after-save flow on Booking__c updates where Is_Canceled__c is true (flows/Booking_Cancellation.flow-meta.xml:161-179). [1]
2. Its first element calls the subflow Issue_Resort_Credit with amount = Cancellation_Credit and the booking's contact (:181-202). The amount is ROUND(Experience Price × Number of Guests × 0.5, 0) (:36-42). [1]
3. The subflow creates a Credit__c record with only Amount__c and Contact__c (flows/Issue_Resort_Credit.flow-meta.xml:28-48). Reason__c is left blank. [1]
4. The rule Large_Credit_Needs_Reason is AND(Amount__c > 500, ISBLANK(Reason__c)) and is active (objects/Credit__c/validationRules/Large_Credit_Needs_Reason.validationRule-meta.xml:4-8). Its own description says Issue Resort Credit does not set a reason. [1]
5. Salesforce documents that record-triggered flows run custom validation rules [2], so the Create Records element fails with FIELD_CUSTOM_VALIDATION_EXCEPTION. Neither the subflow call nor the create element has a fault path, so the error reaches the user instead of being handled. [1]
- 6.

Salesforce's order of execution places after-save record-triggered flows before the final commit [3]. That is consistent with the user seeing 'We can't save this record' and the cancellation not completing.

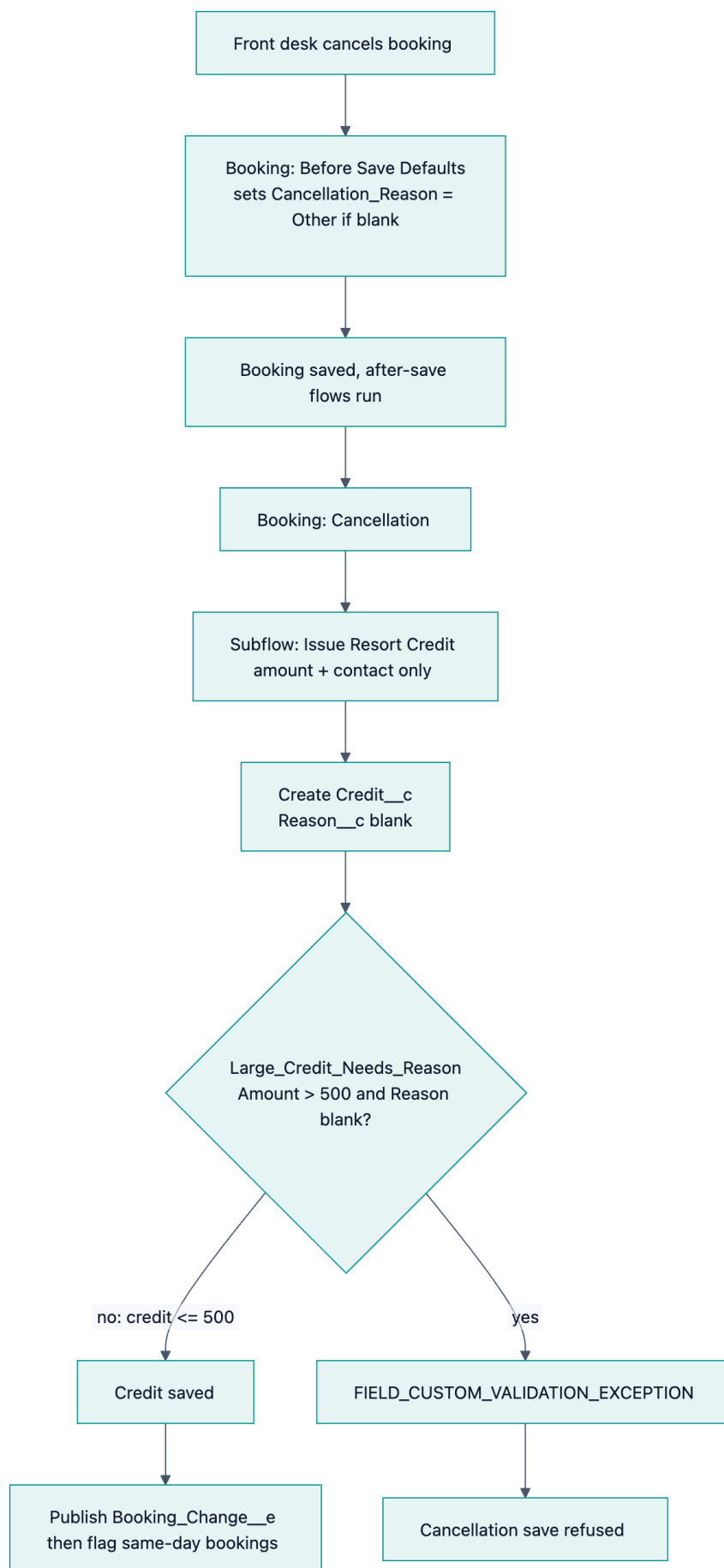
Why only this booking

The credit is half the booking's value, so it exceeds \$500 when price × guests is above roughly \$1,000. For 6 guests that means an experience price above about \$167 per guest. The metadata holds no price for Culinary Heritage Experience (it is record data), so that figure is an implication of the formula, not a confirmed value. [1] Any large-party or high-price cancellation will fail the same way, whatever the experience.

Architecture and dependencies



Booking: Cancellation and the flows it starts. The failing element is Create Credit Record in Issue Resort Credit.



Failure path versus the passing path for small cancellations.

Components involved

Component	Type	Role	Source
Booking: Cancellation	Record-triggered flow (after save, active)	Starts the credit, publishes the event, flags same-day bookings	flows/ Booking_ Cancellation. flow- meta. xml
Issue Resort Credit	Autolaunched flow	Creates the Credit record, no reason	flows/ Issue_ Resort_ Credit. flow- meta. xml:28- 48
Large_Credit_Needs_Reason	Validation rule on Credit__c	Blocks Amount > 500 with blank Reason	objects/ Credit_ _ c/ validationRules/ ...:6
Booking: Before Save Defaults	Record-triggered flow (before save)	Defaults Booking Cancellation Reason to 'Other'	flows/ Booking_ Before_ Save_ Defaults. flow- meta. xml:40- 61
Issue Bulk Resort Credits	Autolaunched flow	Same missing-reason defect	flows/ Issue_ Bulk_ Resort_ Credits. flow- meta. xml:38- 59
Customer Service Assistant agent action	GenAI plugin	Calls Issue Resort Credit	genAiPlugins/ Customer_ Service_ Assistant. genAiPlugin- meta. xml:16

Other automation on Booking

BookingTrigger runs only before insert and before update and checks capacity and quoted price (triggers/BookingTrigger.trigger:1, classes/BookingTriggerHandler.cls:12-28). It does not touch credits. The only validation rule on Credit__c in the workspace is Large_Credit_Needs_Reason. [1]

Assumptions and reproduction

Assumptions

- The workspace metadata matches what runs in the org. The flow definition marks version 1 of Booking: Cancellation active (flowDefinitions/Booking_Cancellation.flowDefinition-meta.xml:3), but that is metadata, not a check of the org. [1]
- Culinary Heritage Experience has a price high enough that 6 guests × price × 0.5 rounds above \$500. Record data was not available.
- Nobody has deactivated or changed the validation rule in the org since this metadata was retrieved.

Steps to reproduce (in a sandbox)

1. Create a booking whose guests × experience price is above about \$1,000, for example 6 guests at \$170.
2. Set `Is_Canceled__c` to true. Expect the error in the report.
3. Repeat with a booking worth under \$1,000. Expect success and a Credit with a blank Reason.
4. To confirm the cause directly, create a Credit with Amount 501 and no Reason. It should fail with the same message.

Remediation tasks

Implementation tasks (for a developer; nothing here was changed)

#	Task	Where	Why / note
1	Add an optional text input reason to Issue Resort Credit and map it to <code>Creditc.Reasonc</code> in Create Credit Record	<code>Issue_Resort_Credit.flow-meta.xml:28-48, :57-73</code>	Fixes the failure. Optional input keeps the AI agent action and other callers working.
2	In Booking: Cancellation, pass a reason into the subflow call, such as text built from <code>Cancellation_Reason__c</code>	<code>Booking_Cancellation.flow-meta.xml:181- 202</code>	Booking: Before Save Defaults already guarantees a reason value ('Other' at minimum), so it will not be blank.
3	Apply the same fix to Issue Bulk Resort Credits: add a reason input and set <code>creditRecord.Reason__c</code>	<code>Issue_Bulk_Resort_Credits.flow-meta.xml:38- 59</code>	Same defect, and it fails for any bulk amount over \$500.
4	Decide the price basis for the credit: current price or <code>Quoted_Price__c</code>	<code>Booking_Cancellation.flow-meta.xml:36- 42;</code> <code>BookingTriggerHandler.cls:32-47</code>	The formula ignores the price the guest was quoted. This is a business decision and it changes credit amounts.
5	Add fault connectors to the subflow call and the create step; route to a visible error or task rather than silently continuing	<code>Booking_Cancellation.flow-meta.xml:181- 202;</code> <code>Issue_Resort_Credit.flow-meta.xml:28- 48</code>	Today any credit failure surfaces as an unhandled flow error to the front desk.
6	Confirm how the agent action passes inputs, and update its instructions if the input list changes	<code>genAiFunctions/ Issue_Resort_Credit/ ; genAiPlugins/ Customer_Service_Assistant.genAiPlugin-meta.xml:16</code>	The agent can issue credits over \$500 and would hit the same rule.

Acceptance criteria

- Canceling a booking worth over \$1,000 completes; a Credit exists with Amount over \$500 and a non-blank Reason.
- Canceling a booking worth under \$1,000 still works; its Credit also has a Reason.
- `Large_Credit_Needs_Reason` remains active and unchanged.
-

Issuing a bulk credit over \$500 through Issue Bulk Resort Credits succeeds when a reason is given.

- If Credit creation fails for any other reason, the user sees a clear message and the booking is not left half-processed.
- Apex or flow tests cover a credit above and below the \$500 line.

What not to do

Warning: Do not deactivate the validation rule or exempt the flow from it. The rule exists so large goodwill credits carry a reason, and the fix is to supply one. A fault path that only swallows the error would let the booking cancel with no credit issued, which is a worse silent failure.

Operational risks and gaps

Risk	Evidence	Impact
Same defect in other credit paths	Issue Bulk Resort Credits and the agent action also omit Reason (Issue_Bulk_Resort_Credits. flow- meta. xml:38- 59)	Failures for large bulk or agent-issued credits
No fault paths in the cancellation chain	Subflow call and Create Credit Record have none; the audit also flags Get Same-Day Bookings and Update Same-Day Booking	Raw platform errors shown to front-desk staff
Credit may not match what the guest paid	Credit uses current Experience price, not Quoted_Price__c (Booking_Cancellation. flow- meta. xml:36- 42)	Over- or under-crediting after a price change; larger amounts hit the \$500 rule sooner
Update inside a loop	Update Same-Day Booking runs once per same-day booking (Booking_Cancellation. flow- meta. xml:151- 160)	DML limit pressure if a guest has many same-day bookings
Front desk cannot create credits directly	Front_Desk permission set gives Credit__c read only (permissionsets/ Front_Desk. permissionset- meta. xml:116- 124)	Not the cause here, since the error is a validation failure, but staff cannot work around it by adding a credit manually

Not verified

Note: Local metadata only. No org runtime evidence was used: the Culinary Heritage Experience price, the booking and credit data, which flow versions are active, and whether the validation rule was edited in the org are all unconfirmed. The root cause does not depend on them: any credit over \$500 without a reason fails this rule, and the flow never sets a reason. Salesforce behavior is cited only to the two documentation passages listed in sources; the Validation Rules help article was found but not read, so nothing is attributed to it.

Sources and Citations

[1] Uploaded Salesforce DX project, demo-salesforce-metadata.zip. internal document provided by the user; file paths and line numbers cited in text are relative to demo/force-app/main/default/.

[2] Record-Triggered Flow Considerations. Salesforce Help.

https://help.salesforce.com/s/articleView?id=platform.flow_considerations_trigger_record.htm&language=en_US&type=5 (accessed October 17, 2026)

[3] Triggers and Order of Execution. Apex Developer Guide, Winter '27 (API 68.0). Salesforce.

https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_triggers_order_of_execution.htm (accessed October 17, 2026)

Data Provenance

Each entry records the request behind a numbered reference: when it was fetched, whether the response was live or cached, and a fingerprint (hash prefix) of the response body, so every cited figure can be re-fetched and verified.

[2] Salesforce Help: Record-Triggered Flow Considerations — fetched 2026-10-17 23:51:51 UTC · served from cache · fingerprint f759e02a4b5753c9

GET salesforce_docs_lookup

[3] Apex Developer Guide: Triggers and Order of Execution — fetched 2026-10-17 23:51:51 UTC · served from cache · fingerprint f759e02a4b5753c9

GET salesforce_docs_lookup