HOUSEBLEND

HOUSEBLEND / RESEARCH & PRACTICAL GUIDANCE

NetSuite Customer Statements: Reconcile Before Sending

> Make NetSuite work better for your finance and operations teams with Houseblend. We help design, implement, integrate and improve ERP systems, with practical support for the people who use them every day.

Published by Houseblend · 2026-11-21 · netsuite customer statements · netsuite statement aging · netsuite consolidated statement · netsuite multi currency statement · accounts receivable reconciliation · statement date vs aging · netsuite generate statements · netsuite ar aging report

Original article: https://www.houseblend.io/articles/netsuite-customer-statements-reconcile-aging-currency



In this article

1. Executive Summary
2. Introduction and Background
3. What a NetSuite Customer Statement Contains and Which Settings Shape It
4. Choosing a Statement Purpose and Setting Combination
5. Date Mechanics: Statement Date, Run Date and Aging As-Of Date
6. Send and No-Send Controls
7. Reconciling Ending Balance, Current Due, Aging and the AR Control Account
8. Multi-Currency, Parent and Child: Testing the Hard Cases
9. Governance: Owner, Exception Queue, Evidence and Automation Handoff
10. Data Analysis and Evidence
11. Implications and Future Directions
12. Frequently Asked Questions (FAQs)
13. Conclusion

Executive Summary

A NetSuite customer statement is a dated document, and a common cause of apparent mismatches between its ending balance, the current amount due and the aging summary is a date difference produced by the generation settings rather than a data error. Oracle documents that a statement lists a customer's invoices, finance charges, credit memos and payments (docs.oracle.com). The **Statement Date** entered on the Generate Statements page is also the date used to calculate aging (docs.oracle.com). The optional **Start Date** decides whether a Balance Forward line appears, and **Show Only Open Transactions** is documented as most useful for statements dated today, with a warning that other dates may produce balance discrepancies (support.jcurvesolutions.com).

The practical consequence is that a statement dated 31 October and reviewed on 12 November describes the account as it stood on 31 October, while a "current amount due" describes it on the review day. A payment received on 5 November explains the gap. A NetSuite consultancy reaches the same conclusion: an ending balance above the amount due means payments have been received since the statement date (limebox.com). A deliberately fictional example (all figures hypothetical) reconciles a $10,500 historical ending balance and a $9,200 current amount due to a zero variance once post-date movement is shown on a sign-off sheet.

Three further design points matter for multi-entity companies. With multiple currencies enabled, each customer's statement carries separate pages for each transaction currency (docs.oracle.com), so a base-currency aging total cannot be compared with one statement page. The **Consolidated Statement** box shows hierarchy balances for parent and child customers, and clearing it shows individual balances (docs.oracle.com). Financial reports run by fiscal period while other reports run by date, a documented source of aging-to-ledger differences (www.citrincooperman.com).

The recommended control has six parts: choose the statement purpose (period-end evidence, current collection request or dispute support); fix the date, start date, open-only, consolidated and currency settings for that purpose; apply send/no-send rules for unapplied payments, credits, write-offs and disputed invoices;

reconcile ending balance, current due, aging and the accounts receivable (AR) control account with date differences shown; test a parent with two children and two currencies in a sandbox; and assign an approval owner, exception queue and retention rule. Reconciling the AR subledger to the general ledger is an established control, and one university procedure sets it as a monthly requirement (onesource.uga.edu). Record-keeping periods are set by tax authorities, for example three, six or seven years in the United States depending on circumstances (www.irs.gov), and six years in Canada (www.canada.ca); confirm applicable periods with a qualified advisor. No benchmark targets are asserted.

Introduction and Background

Readers are usually in the middle of a specific problem. A customer disputes a balance, an AR manager sees a different figure on the aging report, or an administrator is preparing a month-end mass send and wants to know whether the output is safe to release.

This report is written for AR managers, controllers and NetSuite administrators at business-to-business (B2B) and multi-entity companies. It addresses a **monthly statement design and approval control**. It does not cover email automation recipes, advanced PDF template building or dunning cadence design. Formatting is covered in a separate article on Advanced PDF/HTML invoice customization, and scheduled delivery is outside the scope of this report.

A statement of account is generally defined as a detailed report of the contents of an account (www.accountingtools.com), and its traditional purpose is to remind a credit customer of sales that have not yet been paid (www.accountingtools.com). In practice a NetSuite statement serves three different jobs, and separating them is the first design step:

- **Accounting history:** a statement dated at a period end that fixes what the account looked like on that date, useful for audit and dispute support.
- **Collection communication:** a statement dated today that shows what is currently unpaid, useful for asking a customer to pay.
- **Dunning:** a sequenced reminder process. Oracle documents that in customer-level dunning a statement can be attached to the letter, while in invoice and invoice group dunning statements cannot be attached (docs.oracle.com). Statements and dunning letters are separate outputs that can be combined.

The publisher is a Montréal NetSuite consultancy that supports implementation, integration and administration work of this kind. The behaviors described are drawn from Oracle documentation and consultancy publications rather than from a sandbox run, and Oracle's help pages carry no as-of dates, so "as of November 2026" means "as read in November 2026". The send/no-send rules below are proposed policy, not documented NetSuite behavior; the test protocol later in this report shows how to verify them in your own account.

What a NetSuite Customer Statement Contains and Which Settings Shape It

Components of the document

Beyond the transaction list, a statement contains an aging summary at the bottom showing how long each amount has been due (docs.oracle.com). Finance charges are produced by a separate assessment process that Oracle says can run only once per day per customer (docs.oracle.com), so a finance charge invoice dated after the statement date cannot appear on a historical statement. The Standard Statement PDF/HTML template exposes fields for aging buckets and amounts due (docs.oracle.com). Its field list shows two fields both labelled Amount Due (IDs `agingbal` and `amountdue`). This report found no documentation that distinguishes them under every setting combination, so the sign-off process below treats "which Amount Due prints" as a sandbox test item.

A general definition of AR aging is useful here: aging categorizes open customer receivables by the length of time they have been outstanding (www.netsuite.com). Practitioners note that most companies base aging on due date, since it reflects actual payment delinquency rather than invoice age (aaxonix.com).

The settings that change the numbers

The Generate Statements page offers inputs that matter for reconciliation. *Table 1* below maps each setting to its documented effect and to the purpose it best fits.

Setting	What the documentation says	Effect on ending balance, current due and aging	Best fit
Statement Date	The date printed on statements, also used to calculate aging (support.jcurvesolutions.com)	Fixes the "as of" point for aging buckets. A later run date does not move it.	All purposes. Record it on the sign-off sheet.
Start Date (optional)	The oldest transaction shown. With no start date the Balance Forward does not appear (support.jcurvesolutions.com)	A start date creates a Balance Forward line for earlier activity. Without one, long histories appear (www.paidnice.com).	Period-end evidence: set to period start.
Show Only Open Transactions	Most useful for statements printed as of the current date. Other dates may produce balance discrepancies (support.jcurvesolutions.com)	Charge and Payment columns show full original amounts while the Balance column shows what remains (docs.oracle.com), so columns need not foot to the ending balance.	Current collection request, dated today.

Setting	What the documentation says	Effect on ending balance, current due and aging	Best fit
Open Transactions on Statements (accounting preference)	**Open Transactions by Statement Date** includes only transactions open as of the statement date (docs.oracle.com). **Open Transactions by Today** is described by Oracle as including "only invoices with payment dates after the current date" when Show Only Open Transactions is checked; that wording is ambiguous, so test it in a sandbox	Decides whether "open" is judged on the statement date or on today. A consultancy describes the same split (limebox.com).	Period-end evidence: statement-date option. Collection requests: today option.
Consolidated Statement	Shows consolidated balances for customer and subcustomer hierarchies; clearing it shows individual balances (docs.oracle.com)	Changes whose balance is on the page. Parent and child totals differ by design.	Parent-level collection. Clear for child-level disputes.
Include Zero Balances	Adds customers with a zero balance at the statement date who had AR activity since the start date (support.jcurvesolutions.com)	Brings fully paid customers into the batch, which changes how many statements are reviewed.	Period-end evidence batches.

Table 1 shows that no single configuration serves all three purposes. A period-end evidence statement and a collection request are different documents, even for the same customer on the same day. The interpretive point for AR managers is that Open Transactions on Statements is an account-level preference, not a per-run choice. Changing it for one batch changes the meaning of every later batch, so the approval control should record the preference value as well as the run parameters.

A related account-wide preference, **Aging Reports Use**, decides whether reports age by due date or by transaction date. The page reviewed does not say whether it governs the aging summary printed on a statement, so this report does not assert that it does, and the sign-off sheet simply records its value.

The same four inputs (open-only flag, start date, statement date and form number) also drive scripted statement output through the SuiteScript print API (www.netsuiterp.com), and a consultancy describes the same open-only option when a statement is attached to an email (noblue2.com). A script or workflow that generates statements therefore inherits the same date semantics as the manual page.

Choosing a Statement Purpose and Setting Combination

Purpose one: period-end evidence

Period-end evidence answers "what did this customer owe us on the last day of the month?" The statement date is the period end, the start date is the period start, and the "open as of statement date" preference applies. Include Zero Balances is useful because a customer who paid in full on the 30th still had activity in the period. Given the open-only warning for non-current dates, running period-end evidence without Show Only Open Transactions is the conservative design, because charges, payments and credit memos all appear and the ending balance can be proved by arithmetic.

This document is historical. It should not be regenerated later and sent as if it were current. A collections-focused blog makes the generic point that rerunning an older period can reflect closures that happened later (procindex.com), which is why the original PDF should be retained rather than recreated.

Purpose two: current collection request

A collection request answers "what do you owe us today?" The statement date is the run date, Show Only Open Transactions is checked, and the preference is set to Open Transactions by Today (whose Oracle description is ambiguous; confirm its effect in a sandbox). A consultancy recommends this combination for most customer-facing statements because only currently unpaid invoices appear, and the ending balance then equals the amount due (limebox.com). That is an attributed consultancy view, and the test grid below checks it against the reader's own configuration.

Purpose three: dispute support

Dispute support answers "what did the account look like on the date we both remember?" The same consultancy notes that historical statements help resolve payment disputes (limebox.com). A dispute statement must carry its own label (statement date, run date, preference value) and a post-date movement schedule so the customer sees why today's amount due differs.

Date Mechanics: Statement Date, Run Date and Aging As-Of Date

Why several dates exist

The **statement date** is a run input. The **run date** is the day the statement is generated. The sign-off sheet must capture the run date separately because it is what makes post-date movement visible: activity between the statement date and the run date is invisible on a historical statement but present in today's amount due.

Aging reports use their own date logic. An administrator tip on the Oracle community states that the Age column of the A/R Aging Detail report is driven by the "Age As Of Date" field (community.oracle.com). That is a community report and filter labels can vary by version, so record every date filter used on the aging report rather than relying on it as a rule. Oracle also documents that report records are selected by transaction date when filtering by date range and by posting period when filtering by period (docs.oracle.com). One partner guide calls date mismatch the most common reason an AR aging total and the control account disagree (www.business-software.com).

Time zone and date boundaries

Oracle states that the calendar time zone setting affects events and the data returned by reports and searches (docs.oracle.com). The help pages reviewed do not describe how a payment dated at the statement date boundary is treated for a user in a different time zone, so a boundary test belongs in the sandbox: enter a payment late on the last day of the month, run the statement for that date, and note whether the payment appears.

A fictional worked example (Hypothetical Example)

All numbers below are hypothetical and the customer is fictional. The goal is to show how one account produces two different, both correct, numbers.

Customer: Fictional Customer A, all transactions in USD, due-date aging, 30-day terms.

- **Opening position (30 September):** invoice INV-100 for $4,000, dated 5 September, due 5 October.
- **October charges:** INV-101 for $6,000 dated 10 October (due 9 November) and INV-102 for $2,500 dated 20 October (due 19 November).
- **October payment:** $1,500 received 15 October and applied to INV-100, leaving $2,500 open on it.
- **October credit memo:** $500 dated 25 October and applied to INV-102, leaving $2,000 open.
- **Statement parameters:** statement date 31 October 2026, start date 1 October 2026, no open-only filter.
- **Post-date events:** a $2,500 payment on 5 November (applied to INV-100) and a new invoice INV-103 for $1,200 on 10 November.
- **Run date:** 12 November 2026.

The statement dated 31 October shows a Balance Forward of $4,000, charges of $8,500, payments of $1,500 and credit memos of $500, for an ending balance of $10,500. Aging at 31 October, by due date, places $2,500 (INV-100, 26 days past due) in the 1-30 bucket and $8,000 (INV-101 and INV-102) in Current. The buckets sum to the ending balance, so the document is internally consistent.

On 12 November, the amount due is $10,500 less the $2,500 payment plus the $1,200 invoice, or $9,200. The historical ending balance and the current amount due differ by $1,300, and post-date movement fully explains it: a payment after the statement date lowers the amount due, and a new charge raises it. This is the same logic the consultancy cited above gives for ending balances that are higher or lower than the amount due (limebox.com).

If the run had used Show Only Open Transactions, the Charge and Payment columns would show full original amounts and the Balance column the remainder, so INV-100 would show $4,000 as a charge and $2,500 as its balance. A reviewer should add the Balance column, not the Charge column. If the customer says on 12 November that the statement shows $10,500 but the amount owed is $9,200, both are right.

Table 2 below is the customer-level sign-off sheet populated with the example.

Field	Customer A (hypothetical)	Why it is on the sheet
Statement date	2026-10-31	Drives aging and the "as of" point
Run date	2026-11-12	Separates the statement date from today so post-date movement is visible
Opening balance (Balance Forward)	$4,000	Appears only when a start date is entered
Charges	$8,500	INV-101 plus INV-102
Payments	$1,500	Applied to INV-100
Credit memos	$500	Applied to INV-102

Field	Customer A (hypothetical)	Why it is on the sheet
Ending balance	$10,500	$4,000 plus $8,500 less $1,500 less $500
Post-date movement	Payment received $2,500, invoice added $1,200, net decrease $1,300	Events dated after the statement date and on or before the run date
Current amount due	$9,200	$10,500 less $1,300
Aging bucket at statement date	Current $8,000, 1-30 $2,500	Bucket total must equal the ending balance
Currency	USD	One page per transaction currency
Parent grouping	None (standalone)	Records whether the page was consolidated
Variance	$0	Ending balance plus post-date movement minus current due
Owner	Named AR analyst, reviewer sign-off	Maker and checker recorded

As *Table 2* shows, the control is a three-way proof: statement arithmetic, aging bucket total and the post-date bridge to current due. A nonzero variance is an exception to route to the queue, not a reason to adjust figures.

Send and No-Send Controls

A statement is a customer-facing document, so the approval question is not only "is it arithmetically right?" but "should this customer receive it now?" Gaps in the documentation are themselves a reason to test.

Unapplied payments and credits

Practitioners describe unapplied payments as sitting on the customer record as a credit balance (aaxonix.com), and a credit memo as a negative AR transaction that, without explicit application, leaves the aging inflated with offsetting entries (www.snapshotdesign.com). Another reconciliation guide lists credit memos as a source of breaks in two ways (www.business-software.com). The documentation reviewed does not state directly how an unapplied customer payment appears on a generated statement, so this is a mandatory sandbox case. A report listing unapplied transactions is a natural pre-send check.

Customer deposits behave differently. NetSuite's own record documentation states that deposits do not affect the customer's accounts receivable balance (www.netsuite.com), and a consultant notes that deposit balances are not part of the accounts receivable ledger (blog.prolecto.com). A statement can therefore be correct for the receivable and still omit money the customer has already paid.

Write-offs, voids and pending approval

For a bad-debt write-off using a zero-amount payment applied to a write-off journal entry, Oracle states the invoice is no longer included in the Open Invoices report or in A/R Aging reports (docs.oracle.com). Both write-offs and voids change what a post-date statement shows relative to a historical one. On invoice approval,

Oracle states that an invoice needing approval moves to a Pending Approval state (docs.oracle.com). The page reviewed does not say whether a pending invoice appears on statements, so pending-approval invoices should be a named test case.

Disputed invoices

The documentation reviewed does not describe a native dispute flag that excludes an invoice from statements. Many teams add a custom field or saved search to mark disputes. Whatever the mechanism, the send/no-send rule should be explicit, because a disputed invoice that stays on a statement remains part of the legitimate receivable.

Send/no-send rules

- **Hold:** customers with unresolved unapplied payments or credits above the team's own threshold, until the credit is applied or consciously left on account.
- **Hold:** customers with a Pending Approval invoice that the sandbox shows appears on the statement.
- **Hold:** customers whose write-off or void was entered but not yet approved.
- **Send with note:** customers with a disputed invoice, where the statement carries a line identifying it as under review.
- **Send:** customers whose three-way proof shows zero variance.
- **Do not send:** customers without an email address on the customer record, since each customer needs one on the record to be emailed (www.paidnice.com).

Any numeric thresholds in the first rule are the organization's own policy choices, and no public benchmark for an acceptable unapplied-credit level was found.

Reconciling Ending Balance, Current Due, Aging and the AR Control Account

Four numbers, four definitions

An AR manager meets four numbers that look alike:

- **Ending balance:** the arithmetic result on the statement at the statement date.
- **Current amount due:** what is open on the review day, per a statement run today or the customer record.
- **Aging total:** the sum of the buckets at the aging as-of date.
- **AR control account:** the general ledger (GL) balance of the receivable account, in base currency.

Several reports can present these numbers, each with its own date and period logic, so a difference between them does not by itself indicate wrong data.

Why aging and the GL can differ

Consultancy sources describe recurring causes, presented here as general reconciliation reasoning:

- **Period versus date filtering:** financial reports such as the balance sheet run by fiscal period while other reports run by date (www.citrincooperman.com). The same firm explains that the aging report captures data based on the transaction dates (www.citrincooperman.com), and reports that changing a per-user Report by Period setting made the aging report match the balance sheet in its example (www.citrincooperman.com).
- **Journal entries to AR without a customer:** a consultancy notes that such entries risk a reconciliation issue between the trial balance and the aging report (optimaldataconsulting.com) and recommends avoiding journal entries posted directly to AR and AP accounts (optimaldataconsulting.com).
- **Unapplied balances:** an aging review article notes that totals differ by the unapplied balance (procindex.com).
- **Deposits and credits:** deposits sit outside the receivable (www.netsuite.com), and unapplied credit memos leave offsetting entries (www.snapshotdesign.com).

A reconciliation order

1. Prove the statement: opening balance plus charges less payments less credit memos equals ending balance.
2. Prove the buckets: aging summary total equals the ending balance at the same statement date.
3. Bridge to today: ending balance plus post-date movement equals current amount due.
4. Tie to the aging report, run with matching date, currency and subsidiary filters, and record the filters.
5. Tie the base-currency aging total to the AR control account GL balance, listing reconciling items (journal entries with no customer, period mismatches, revaluation, deposits).

Steps 1 to 3 belong to the statement approver. Steps 4 and 5 belong to the controller or an accounting reviewer. Published public-sector procedures show the expected shape of step 5: California's State Administrative Manual directs agencies to reconcile detailed accounts receivable subsidiary amounts with the general ledger control totals (www.dgs.ca.gov) and to prepare AR reconciliations monthly within 30 days of the preceding month (www.dgs.ca.gov). Those figures are one public body's rule, not a benchmark for private companies, and the close calendar should be set by the company's own policy.

Multi-Currency, Parent and Child: Testing the Hard Cases

Currency behavior

With Multiple Currencies enabled, Oracle states that statements include separate pages for each transaction currency for each customer (docs.oracle.com). A vendor blog notes the standard template has no multi-currency support (www.paidnice.com), and a specialized multi-currency statement template exists. The ledger works differently: transactions post to the general ledger in the base currency (docs.oracle.com). The reconciliation implication is that a USD page and a EUR page of one statement must be reconciled in their own currencies first and only then converted for comparison with base-currency aging or the GL. Summing page totals of different currencies is a review-process error, not a NetSuite error.

Two ledger events then move base-currency values without moving the statement. Oracle states that revaluations apply exchange rates as of the last day of the accounting period to calculate gain or loss (docs.oracle.com), and practitioners describe the process as posting the total variance to the Unrealized Gain/Loss account (www.kolleno.com), with multi-currency AR balances revalued at period end (aaxonix.com). Accounting standards describe the same concepts. The IASB defines the closing rate as the spot exchange rate at the end of the reporting period (www.ifrs.org), a Deloitte IAS Plus summary says monetary items are retranslated using the closing rate (www.iasplus.com), and FASB Statement 52, the pre-Codification standard behind ASC 830, requires recorded foreign-currency balances to be adjusted to reflect the current exchange rate at each balance sheet date (storage.fasb.org). Exchange differences arising are recognised in profit or loss (www.icaew.com).

A hypothetical illustration: a EUR 1,500 invoice booked at 1.10 is $1,650 in base currency. If period-end revaluation uses 1.12, the receivable is $1,680 and a $30 unrealized gain posts, while the customer's EUR statement still shows EUR 1,500.

Parent and child behavior

NetSuite's schema documentation describes a consolidated balance as the current accounts receivable balance due for the customer-subcustomer hierarchy (www.netsuite.com), and Oracle states that with Consolidated Payments the A/R Aging report shows the un-consolidated balance for each subcustomer (docs.oracle.com). A partner guide notes each subcustomer remains a customer record with its own fields and transaction history (versich.com). For statements, the Consolidated Statement box chooses between hierarchy balances and individual balances. One vendor source states that consolidated statements have no schedule of their own (www.paidnice.com), which matters for the automation handoff.

NetSuite's consolidated invoicing SuiteSolution is a different feature from the Consolidated Statement option: its terms state that consolidated invoices are virtual invoices and the general ledger impact remains with the individual invoices (www.netsuite.com). In OneWorld accounts, a multi-subsidiary customer keeps a balance per subsidiary, so the test should also vary the subsidiary where relevant.

Test-case grid

Table 3 lists the five required test cases. The "documented basis" column states what sources say, and the last column states what the tester should verify in a sandbox. No cell asserts a result that was not documented.

Test case	Setup (hypothetical)	Documented basis	Result to confirm in sandbox
No start date	Customer with 14 months of history. Run with start date blank.	No Balance Forward and full history (www.paidnice.com)	Page length, absence of Balance Forward line, ending balance equals aging total
Open-only, historical date	Statement date 31 Oct, run 12 Nov, open-only checked, both preference values	Other dates may produce discrepancies. The preference decides "open"	Which invoices appear, how Charge, Payment and Balance columns behave, whether ending balance equals aging at 31 Oct

Test case	Setup (hypothetical)	Documented basis	Result to confirm in sandbox
Multiple currencies	Parent with USD and EUR invoices, one EUR payment	Separate pages by transaction currency	Page count, currency of each balance, aging buckets per page, which Amount Due field prints
Consolidated child	Parent with Child A (USD and EUR) and Child B (USD and EUR). Run once consolidated, once cleared	Hierarchy balance versus individual balance	Parent page equals sum of child balances per currency. Aging report shows un-consolidated child balances
Unapplied credit	Child B has a $500 unapplied payment and a $300 unapplied credit memo	Unapplied payments sit as credit balance (aaxonix.com). Deposits excluded from AR (www.netsuite.com)	Whether credits reduce ending balance, how they appear in aging, and the difference from a deposit

Table 3 is a protocol, not a result set. Its value is that every row names an observable output and a source of expected behavior, so a failed row identifies a configuration decision rather than a vague "mismatch".

A recommended parent test: Parent P has Child A and Child B. Child A has a USD invoice of $3,000 and a EUR invoice of EUR 2,000. Child B has a USD invoice of $1,000 and a EUR invoice of EUR 1,500 with a EUR 500 unapplied payment. Expected consolidated totals, subject to sandbox confirmation, are USD $4,000 and EUR 3,500 gross, or EUR 3,000 if the unapplied EUR 500 nets against the balance. Run consolidated and individual versions and compare per currency.

Governance: Owner, Exception Queue, Evidence and Automation Handoff

Approval owner and permissions

Oracle documents a "Generate Statements" permission (ID `TRAN_STATEMENT`) with levels None, View, Create, Edit and Full (docs.oracle.com). Whether preparer and approver differ is a control design choice, but published guidance is consistent. Stanford's Administrative Guide says the same individual should not prepare and also review a reconciliation (adminguide.stanford.edu), and a University of Texas System policy says that where one person must perform all functions the department must assign a second person to review the work (www.utsystem.edu). NIST SP 800-53 control AC-5 frames the system side, requiring system access authorizations that support separation of duties (nvlpubs.nist.gov). For small AR teams, the PCAOB's internal control standard recognizes that limited staff can restrict segregation of duties, leading a company to implement alternative controls (pcaobus.org). A documented compensating review, such as a controller sampling sign-off sheets, is the practical answer.

Exception queue

An exception queue is a saved list of customers that failed a proof step or matched a hold rule. Each entry needs a reason code (variance, unapplied credit, pending approval, write-off pending, dispute, missing email, currency mismatch), an owner and a release decision. A University of Miami reconciliation procedure illustrates

the review flow, with reconciliations submitted to a cost center manager for review (controller.miami.edu). Resolved exceptions should update the sign-off sheet, not just the customer record.

Evidence retention

Retain, for each run: the parameters (statement date, start date, open-only, consolidated, zero balances, the preference value and the Aging Reports Use value), the generated PDFs, the sign-off sheet and the exception disposition. A University of Georgia procedure similarly requires evidence of who prepared or reviewed the reconciliation and when (onesource.uga.edu). NetSuite's System Notes track record changes and support the audit trail. One workflow author states that NetSuite generates the statement PDF dynamically at send time (theerpguy.me), which reinforces keeping the original PDF as evidence. Whether a mass-sent statement leaves a message record on the customer should be confirmed in the sandbox.

Retention periods are set by tax authorities and vary by circumstance. The IRS lists three, six and seven years for different situations (www.irs.gov), and the Canada Revenue Agency states six years from the end of the last tax year (www.canada.ca). Whether customer statements fall within the records a given company must keep is a question for its tax or legal advisor.

Integrations and source data

An integration can change what a statement shows, because the statement reflects whatever lands in NetSuite. A vendor template documents that a connected storefront integration creates NetSuite customer payments against corresponding invoices (docs.celigo.com) and syncs companies to NetSuite as customers (docs.celigo.com). The control consequence is a cut-off: integration batches that post payments or customers should be complete and reviewed before the statement run, and the run time recorded.

Automation handoff

Once the manual control works, scheduling can be added. Practitioners describe a Send Email action with Include Statement checked (theerpguy.me), and users describe workflows for sending statements (netsuitehub.com). A dunning-template guide describes attaching a statement with its own open-transactions checkbox (www.kolleno.com). Mass send works from a customer list, so the sign-off sheet should be complete before the list is selected. Automation should follow an approved sign-off and never replace it. Scheduled delivery and reminder sequencing are covered in the dunning-engine article when it is published.

Data Analysis and Evidence

No public benchmark for statement mismatch rates, exception rates or review times was found, and no target is asserted here. The quantitative evidence available is documentary and consists of parameters.

Documented numeric parameters

- **Five permission levels** for the Generate Statements permission: None, View, Create, Edit and Full.
- **One finance-charge assessment per customer per day.**

- **Three, six and seven year** US record periods depending on situation, and **six years** in Canada from the end of the last tax year. The US periods are (www.irs.gov), and the Canadian period is (www.canada.ca).
- **Monthly reconciliation, within 30 days of the preceding month** in one state government's AR procedure (www.dgs.ca.gov).

Hypothetical worked arithmetic: the $10,500 ending balance at 31 October, less a $2,500 post-date payment, plus a $1,200 post-date invoice, gives a $9,200 current amount due at 12 November and a $0 variance after the bridge.

The statement-versus-today gap in the example is 12.4% of the historical ending balance ($1,300 divided by $10,500). That percentage is an artifact of invented numbers and is not a benchmark.

Accounting concepts support the date discipline. IAS 10 defines events after the reporting period as those occurring between the end of the reporting period and the date the financial statements are authorized for issue (www.ifrs.org), and divides them into those providing evidence of conditions at period end (www.ifrs.org) and those indicating conditions that arose afterwards (www.ifrs.org). In audit practice, post year-end cash testing is described as a way to assess whether a receivable balance is recoverable (www.accaglobal.com), and cash received before year end but still in receivables is a cut-off issue (www.accaglobal.com). These concepts govern financial statements and audits, not customer statements, but they show the same separation between a fixed period-end position and later events.

Implications and Future Directions

Three implications follow. First, the approval control is mostly a date-labeling control: most mismatches above vanish when the statement date, run date and post-date movement sit on one sheet. Second, account-level preferences are shared state, so a change made for one run changes later runs and belongs in the change log. Third, multi-currency and hierarchy cases are where documentation is thinnest and sandbox testing most valuable.

Questions to answer per account rather than assume: whether pending-approval invoices appear, how unapplied payments display, which Amount Due field prints, how boundary-date payments are treated across time zones, and whether mass-sent statements leave a message record. A NetSuite administrator or implementation partner can run the test grid in a sandbox, and the grid should be re-run after major releases.

Frequently Asked Questions (FAQs)

How do I generate customer statements in NetSuite?

Oracle documents the Generate Statements page under Customers, Accounts Receivable. Enter the statement date, an optional start date and the open-only, consolidated and zero-balance options, then select customers and print or email. Customers need an email address to be emailed.

Why does the customer statement aging not match the aging report?

Common documented reasons are different dates (the statement date versus report filters, including the Age As Of Date in the detail report (community.oracle.com)), date-range versus period selection, and base-currency reports versus transaction-currency statement pages.

What is the difference between a statement date and an aging as-of date?

The statement date is the generation input and drives aging on the statement. The aging report has its own filters, so record both and use the same date when comparing.

Why is my statement ending balance different from the amount due?

If the statement is historical, activity after the statement date (payments, credits, new invoices) explains the difference, as in the worked example. A consultancy ties the statement-date preference value to historical views and the "today" value to current ones.

How do multi-currency customers appear on statements?

As separate pages per transaction currency, with each page in its own currency. Reconcile each currency separately before converting to base.

How do parent and child customers appear, and what is the consolidated balance?

With the Consolidated Statement box checked, hierarchy balances appear. Cleared, each customer's individual balance appears. The consolidated balance field describes the hierarchy's current balance due (www.netsuite.com).

Can I mass send customer statements?

Yes. On Customers > Accounts Receivable > Generate Statements, select the customers to receive statements, then choose Email from the Actions list; each customer needs an email address on the customer record (docs.oracle.com). Oracle also advises submitting each page of the customer list separately, because information is not saved when you switch between pages. Run the sign-off sheet before the send, not after.

Conclusion

A NetSuite customer statement is safe to send when its purpose, date, balance view and currency treatment are known and recorded. The documented settings explain why a historical ending balance, a current amount due and an aging total can all be correct at once: the Statement Date fixes aging, open-only is designed for statements dated today, consolidated statements change whose balance is shown, and multi-currency customers receive a page per transaction currency.

The control proposed here is small: a purpose label, a recorded parameter set, a three-way proof, a send/no-send rule set, a five-step reconciliation to the AR control account, a sandbox test grid and a named owner with retention. Each element can be adopted independently. Teams that need help designing the sandbox tests,

building saved searches for the exception queue or connecting the process to integrations can engage a NetSuite administrator or consultancy, and the rules above should be adapted to the company's own policies, auditors and tax advisors.

Source links

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

1. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/section_N1300430.html
2. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/section_N1300518.html
3. https://support.jcurvesolutions.com/blog/generating-statements
4. https://limebox.com/why-your-netsuite-customer-statement-balances-dont-match-and-how-to-fix-it
5. https://www.citrincooperman.com/In-Focus-Resource-Center/NetSuite-Report-Troubleshooting-Reconciling-Aging-Reports-to-the-Balance-Sheet
6. http://onesource.uga.edu/wp-content/uploads/financial_review_procedures.pdf
7. https://www.irs.gov/businesses/small-businesses-self-employed/how-long-should-i-keep-records
8. https://www.canada.ca/en/revenue-agency/services/tax/businesses/topics/keeping-records/where-keep-your-records-long-request-permission-destroy-them-early.html
9. https://www.accountingtools.com/articles/what-is-a-statement-of-account.html
10. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/section_4043740474.html
11. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/section_N1301640.html
12. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/SBADVTemplates_1863265063.html
13. https://www.netsuite.com/portal/resource/articles/accounting/accounts-receivable-aging.shtml
14. https://aaxonix.com/resources/netsuite-accounts-receivable
15. https://www.paidnice.com/blog/netsuite-customer-statements
16. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/section_N1385293.html
17. http://www.netsuiterp.com/2019/01/print-transactions-and-statements-using.html
18. https://noblue2.com/blog/netsuite-tips-sending-emails-from-netsuite
19. https://procindex.com/blog/collections-aging-review
20. https://community.oracle.com/netsuite/english/discussion/4476204/netsuite-admin-tips-a-r-aging-detail-computation-of-age-column
21. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/article_1210032559.html
22. https://www.business-software.com/blog/accounts-receivable-reconciliation
23. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/section_N500123.html
24. https://www.snapshotdesign.com/snapshot-blog/reconciling-b2b-billing-workflows-netsuite
25. https://www.netsuite.com/help/helpcenter/en_US/srbrowser/Browser2017_2/schema/record/customer.html
26. https://blog.prolecto.com/2020/08/01/learn-the-pattern-to-write-off-netsuite-customer-balances
27. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/bridgehead_N1481661.html
28. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/section_4171578435.html

29. https://optimaldataconsulting.com/resources/how-to-fix-the-no-customer-vendor-line-on-netsuite-aging-report
30. https://www.dgs.ca.gov/Resources/SAM/TOC/8200/8294
31. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/section_N1397082.html
32. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/section_N1430227.html
33. https://www.kolleno.com/revalue-open-currency-balances-in-netsuite
34. https://www.ifrs.org/content/dam/ifrs/project/lack-of-exchangeability-amendments-to-ias-21/ed2021-4-lack-of-exchangeability-ias-21.pdf
35. https://www.iasplus.com/en/standards/ias/ias21
36. https://storage.fasb.org/aop_fas52.pdf
37. https://www.icaew.com/technical/corporate-reporting/ifrs/ifrs-accounting-standards-tracker/ias-21-the-effects-of-changes-in-foreign-exchange-rates
38. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/section_N1288474.html
39. https://versich.com/blog/netsuite-parent-child-relationship-setup-for-customer-hierarchies
40. https://www.paidnice.com/blog/netsuite-oneworld-collections
41. https://www.netsuite.com/portal/assets/pdf/suitesolution-consolidated-invoicing-tos-v010117.pdf
42. https://docs.oracle.com/en/cloud/saas/netsuite/ns-online-help/chapter_N3236764.html
43. https://adminguide.stanford.edu/chapters/financial-administration/account-balance-reconciliation-and-attestation/account-balance
44. https://www.utsystem.edu/sites/policy-library/policies/hop-217-cost-centerproject-reconciliations-segregation-of-duties
45. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf
46. https://pcaobus.org/oversight/standards/auditing-standards/details/AS2201
47. https://controller.miami.edu/_assets/pdf/documents/balance_sheet_reconciliation_presentation.pdf
48. https://theerpguy.me/blog/netsuite-customer-statement-email-workflow
49. https://docs.celigo.com/hc/en-us/articles/39525523971995-Shopify-B2B-NetSuite-integration-template
50. https://netsuitehub.com/forums/topic/sending-statements-to-customers
51. https://www.kolleno.com/attach-statements-to-dunning-templates
52. https://www.ifrs.org/issued-standards/list-of-standards/ias-10-events-after-the-reporting-period
53. https://www.accaglobal.com/content/dam/acca/global/PDF-students/acca/f8/specimen/aa-specimen-S16%20.pdf

THE PUBLISHER

About Houseblend

Houseblend is a NetSuite consulting firm serving finance and operations teams. We help organizations implement ERP systems, connect business applications, improve existing configurations and maintain the systems that support everyday work. Our audience includes finance leaders, controllers, operations managers, NetSuite administrators and implementation teams.

Implementation and architecture

Houseblend provides NetSuite implementation, architecture and data migration services. We help teams evaluate how business processes, reporting requirements and existing data should fit together in an ERP environment. Training supports the people responsible for adopting and operating the resulting system.

Integrations, customization and AI

Our services include NetSuite integrations and customization, as well as AI integrations and AI transformation work. These engagements connect ERP data and workflows with the broader application landscape. The right design depends on the organization's systems, controls and operating needs.

Improve and support an existing system

Houseblend offers NetSuite health checks, optimization, managed support and project rescue services. We also provide expertise for analytics and specialist workflows, including NetSuite Analytics Warehouse, warehouse management and field service management. Published educational material helps teams investigate options and prepare informed questions for their implementation or support work.

Work with Houseblend

Explore NetSuite implementation, integrations, managed support and AI integrations. Contact Houseblend to discuss your current system and priorities.

Article examples explain concepts rather than promising a particular license, product capability, delivery schedule or outcome. Engagement scope is confirmed with the Houseblend team.

Website: https://www.houseblend.io

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.