NetSuite Statistical Accounts: Overview, Role, and Function

By Houseblend Published June 13, 2025 15 min read

NetSuite Statistical Accounts – Overview and Role

NetSuite **Statistical Accounts** are special ledger accounts (visible in the Chart of Accounts) that track non-financial quantities without affecting the General Ledger (Source: docs.oracle.com) (Source: technologyblog.rsmus.com). Introduced in NetSuite's Advanced Financial module, these accounts let finance teams capture metrics like employee headcount, square footage, or issued shares and use them in financial reports and analysis. Unlike regular accounts, statistical accounts are *always debit-positive* and **do not post to the GL** (Source: docs.oracle.com)(Source: technologyblog.rsmus.com); instead, their balances exist purely for reporting, budgeting, and

allocation purposes. Once defined (with a unit of measure and optional segments like Department or Class), statistical accounts can feed into custom reports and financial ratios. For example, a "Headcount" stat account might supply the divisor in an Earnings-Per-Share (EPS) calculation, while a "Square Footage" stat account could drive cost allocations of rent or utilities (Source: docs.oracle.com)(Source: houseblend.io).

Statistical accounts have three primary roles in NetSuite: tracking *external* metrics (e.g. office space, issued shares), tracking *internal* metrics (e.g. employee count, inventory units), and serving as **weights in allocation schedules** (Source: docs.oracle.com). All values recorded in a stat account (via statistical journal entries) can be reviewed or automated through saved searches and schedules. These values can then be merged into financial reports: for instance, custom income statements can display statistical balances alongside revenues and expenses, or compute ratios (e.g. Net Income ÷ Share Count) (Source: docs.oracle.com)(Source: netsuite.folio3.com). In short, statistical accounts provide a bridge between operational data and financial analysis, helping organizations include key non-financial data in budgeting, reporting, and KPI calculations (Source: houseblend.io)(Source: netsuite.folio3.com).

Tracking Non-Financial Metrics and KPIs

By design, statistical accounts let organizations incorporate operational and non-monetary data into financial reporting. Finance teams track metrics such as **square footage**, **headcount**, **units sold**, **support tickets**, or **machine hours** as statistical balances. These metrics can then be used in income statements, ratio analyses, or expense allocations. For example, NetSuite's documentation illustrates using a square-footage stat account to allocate rent or office expense by office area (Source: docs.oracle.com)(Source: docs.oracle.com). Another example is a "Shares Outstanding" stat account feeding into the EPS calculation. In practice, any recurring count or measure can be tracked: Folio3 notes that statistical accounts are ideal for KPIs outside of revenue/expense, such as *support tickets per day*, *average order value*, or *employee turnover rate* (Source: netsuite.folio3.com). They also enable operational analyses: manufacturing firms might track production output or **inventory turns**, while service businesses might track **calls or service hours**.

Because statistical accounts do not affect financial totals, they give management "insight into parts of the business that would otherwise be invisible" (Source: houseblend.io)(Source: netsuite.folio3.com). For instance, a nonprofit could track **volunteer hours** or **donors** and allocate fundraising costs by those counts. A SaaS company might track **active subscriptions** or **support incidents** to allocate customer service costs or measure churn. In general, any metric that has a relationship to costs or revenues can be statted and then analyzed. NetSuite even lets you budget

on stat accounts: you can create Budgets for a stat account (e.g. budget 100 new hires in Sales next year) and compare *Budget vs. Actual* (Source: docs.oracle.com). This tight integration with reporting makes statistical accounts powerful for KPI tracking and data-driven decision-making (Source: netsuite.folio3.com).

Configuration and Setup

Enable Features: Before using statistical accounts, the Statistical Accounts feature must be enabled under **Setup > Company > Enable Features > Accounting > Advanced**. In practice you will also check **Expense Allocation** and **Dynamic Allocation** at the same time, since statistical accounts are often used with allocation schedules (Source: technologyblog.rsmus.com)(Source: netsuite.folio3.com). An example Enable Features page is shown below, with **Statistical Accounts**, **Dynamic Allocation**, and **Expense Allocation** selected.

NetSuite > Setup > Company > Enable Features > Accounting: enable "Statistical Accounts" (and usually "Dynamic Allocation" and "Expense Allocation") (Source: technologyblog.rsmus.com) (Source: netsuite.folio3.com).

Units of Measure: Because statistical accounts track quantities, each stat account is tied to a **unit type**. NetSuite uses the Units of Measure feature (Lists > Accounting > Units of Measure) to define custom units. For example, you might create a unit type **Area** with units like *Square Footage (SQFT)*, or a unit type **Headcount** (with base unit 1). When creating a stat account, you *select* an existing unit type or click **New** to create one on the fly (Source: docs.oracle.com)(Source: technologyblog.rsmus.com). (The base unit within the type is fixed once saved.) For example, Concentrus illustrates defining an **Area** type and adding a unit "Square Footage (SQFT)" (Source: blog.concentrus.com), while RSM's example creates a **Headcount** unit type for counting employees (Source: technologyblog.rsmus.com). NetSuite will only allow one unit type per stat account; all journal entries against that account must use the same unit of measure (Source: docs.oracle.com).

Creating a Statistical Account: Once units are set up, create a stat account as follows (similar to other accounts):

1. **Go to** *Setup > Accounting > Manage G/L > Chart of Accounts > New*.
2. **Select Type =** *Statistical*. Enter an **Account Number** (if used) and a descriptive **Name** (e.g. *Headcount*, *Office SqFt*).
3. **Unit Type:** Select the unit type defined above (the default unit will auto-fill). (Note: currency fields are hidden since stat accounts are non-monetary (Source: docs.oracle.com).)

4. **Description & Date:** Optionally add a description and set the effective date (defaults to today) for when the account can start receiving entries.

5. **Segmentation:** If you use Departments, Classes, or Locations, you can *restrict* the stat account to a single segment. For example, if you only want a "Facility SqFt" stat account to apply to Building A, select Department=Buildings. Important: once you lock in a Department/Class/Location on the stat account, any statistical journal entry must use that same segment (it cannot be used in entries for other departments) (Source: docs.oracle.com).

6. **Subsidiaries (OneWorld):** In NetSuite OneWorld, a stat account can be assigned to multiple subsidiaries. Any balances from those subs roll up into the stat account (with an exchange rate of 1:1, since stat accounts are not currency-converted) (Source: docs.oracle.com). Check **Include Children** to automatically include any child subsidiaries of the ones selected (Source: docs.oracle.com).

When finished, click **Save**. You will now see the account in your Chart of Accounts (note it is always debit balance by definition (Source: netsuite.folio3.com)). At this point the account is ready for statistical journal entries or scheduling.

Populating Statistical Accounts

Statistical accounts are maintained via **Statistical Journal Entries** or **Statistical Schedules**. A statistical journal entry is a one-sided entry (only affecting stat accounts) by Dept/Class/Location. Entries can be **manually created** (Transactions > Financial > Make Statistical Journal Entries > New) or **automated** via scheduled saved searches.

- **Manual Journals:** You can enter the change (delta) in the stat account or post an **absolute value**. For example, if a monthly report says *Office Space = 10,000 sq ft per department*, instead of recording +500 vs last month, you can post an absolute balance journal that "overwrites" the stat account to 10,000 (Source: docs.oracle.com). (NetSuite calls this an *Absolute Balance Update Offset Journal*.) This is useful for periodic metrics like total headcount or total sq ft, so each month's entry shows the point-in-time total (Source: docs.oracle.com). Note that NetSuite's system date is used on such offset journals (Source: docs.oracle.com). All entries – whether incremental or absolute – appear in the *Statistical Account Register*, which you can view by clicking the account name in the Chart of Accounts (Source: docs.oracle.com).

- **Scheduled Journals:** Alternatively, set up a **Statistical Schedule** to auto-generate entries. First build a saved search that returns the values you want (for example, an Employee saved search grouped by Department, with **Summary Type = Count** on *Name* to count employees per dept (Source: sikich.com)). Then go to *Transactions > Financial > Create Statistical Schedule*. Link your Stat Account and the saved search, choose frequency (Monthly, Quarterly, etc.) and start date (Source: sikich.com). When you **Run** or **Create Journal Entry** from that schedule, NetSuite will create one stat journal line per saved-search row, updating the stat account. Sikich's guide illustrates this process for a "Headcount" account by department (Source: sikich.com)(Source: sikich.com). Make sure your saved search's results columns match the stat account's unit (e.g. a count for headcount) (Source: sikich.com).

Because of the options above, a key best practice is to **understand how the stat balance is maintained**. As a consultant noted, stat accounts can either accumulate deltas or hold only the latest value (Source: reddit.com)(Source: docs.oracle.com). If you intend to take "snapshots" each period, use absolute entries (or an offset journal) so the stat account always shows the end-of-period total. If you want a running total of all changes, use regular stat journals without resetting. Plan this in advance: e.g. if using deltas, your KPI formulas must sum all entries; if using snapshots, use the most recent entry.

Integration with GL and Allocations

Although statistical accounts themselves do **not post to the GL**, they integrate closely with NetSuite's allocation and reporting tools. The **Dynamic Allocation** feature lets you use a stat account's balance as the weighting factor for expense allocations (Source: docs.oracle.com). For example, to allocate a department's rent expense by headcount, you would: define a *Headcount* stat account (with up-to-date counts per department), then create an Allocation Schedule that uses that stat account's balance as the allocation weight. When the schedule runs, NetSuite generates real GL journal entries moving expense amounts proportionally across subsidiaries, depts, or classes. As Oracle notes, if Dynamic Allocation is enabled, *"users can assign any statistical account to an allocation schedule. The weight for the allocation…is dynamically calculated at the time the allocation journal is generated"* (Source: docs.oracle.com). The step-down allocation feature even allows chaining multiple schedules, treating one's result as the input to the next (Source: docs.oracle.com).

Stat accounts also support **budgeting**. You can enter budget amounts for a stat account by segment (e.g. budget 50 new hires for Dept A), and then include that in Budget vs Actual reports (Source: docs.oracle.com). In one example, a company budgeted the number of employee

requisitions in a *Headcount* stat account and then compared it to actual hires (Source: docs.oracle.com). Thus, non-financial budgets (like expected meters, counts, etc.) can be tracked alongside financial budgets.

On the reporting side, once stat accounts have values, you can customize financial statements to display them. For instance, you might add a column showing *SQFT* next to an Income Statement, or compute and show ratios (Net Income ÷ stat balance) such as EPS (Source: docs.oracle.com) (Source: netsuite.folio3.com). In short, statistical accounts feed into the GL via allocation journals and reporting, even though they themselves carry no monetary value in the ledger (Source: docs.oracle.com)(Source: technologyblog.rsmus.com).

Example Use Cases and Industries

NetSuite statistical accounts are extremely versatile. Common use cases include:

- **Expense Allocation (Headcount):** Create a *Headcount* stat account that tracks number of employees in each department or subsidiary (via saved search or HR integration). Use it to allocate salaries, utilities or rent by headcount. *Example:* Sikich shows building a stat schedule to count employees by dept and then using that in dynamic expense allocation (Source: sikich.com)(Source: sikich.com). Similarly, RSM describes allocating office costs across departments using headcount weights (Source: technologyblog.rsmus.com).

- **Space/Area Metrics:** Track *square footage* or *capacity* by location or department. For instance, record each office's occupied square feet monthly. Then allocate rent or maintenance expenses by those sq ft values. NetSuite's official docs specifically note using a stat account to track office space by department as an example (Source: docs.oracle.com)(Source: docs.oracle.com). Folio3 and Concentrus similarly cite office-square-footage and rent-per-area as textbook uses (Source: docs.oracle.com)(Source: docs.oracle.com).

- **Usage and Volume Metrics:** In manufacturing or service companies, track operational KPIs as stat accounts. Examples include **production volume** (units produced per line), **machine hours**, **customer support tickets**, or **transactions processed**. Folio3 points out using stat accounts for any KPI outside normal revenues/expenses – e.g. support tickets per day or average order value (Source: netsuite.folio3.com). A Windows-and-Doors manufacturer might track *square feet of material* used by product line, then compute profitability per area (Source: houseblend.io).

- **Financial KPIs (EPS, etc.):** Use stat accounts for financial ratios. A classic example is *Outstanding Shares*. Enter the count of issued shares in a stat account, and then your income statement can show EPS = Net Income ÷ Shares (Source: docs.oracle.com). Nonprofits might analogously track metrics like *number of program participants* or *volunteer hours* to analyze cost per participant. SaaS firms could track *active subscriptions* or *support cases* to allocate support costs. (While specific industry examples beyond those cited are scarce in NetSuite docs, the general principle is that any measurable quantity can be statted and reported or allocated.)

Industry Highlights:

- *Nonprofits:* Common non-financial drivers include volunteer hours, program beneficiaries, or fundraising targets. A nonprofit might track "Meals Served" or "Students Trained" in stat accounts to allocate overhead or report program efficiency (though no NetSuite document explicitly covers this, it follows the same logic of operational metrics).

- *SaaS/Technology:* These companies often use statistical accounts for metrics like **number of user licenses**, **API calls**, or **churn count**. For example, a SaaS could track monthly active users per customer and use that to allocate support costs or compute revenue per user. Folio3's mention of *support tickets* and *average order value* (Source: netsuite.folio3.com) applies directly here.

- *Manufacturing:* Besides material usage, manufacturers can track **production hours** or **yield rates**. Houseblend's example of an income statement by material area (Source: houseblend.io) illustrates this class of use. Also, production output counts can feed into variance analyses or resource allocation studies.

These examples show that **any industry** can adapt stat accounts for its key metrics. NetSuite customer blogs and consultants frequently emphasize that *"no matter your business model, you can have NetSuite statistical accounts be part of your business reporting"* (Source: houseblend.io).

Best Practices for Maintenance and Audit

To keep statistical accounts reliable and auditable:

- **Clear Naming and Documentation:** Give each stat account a descriptive name (including units if helpful, e.g. *"OfficeSqFt – City HQ"*) and description. Document the purpose, unit of measure, and any segments in an internal guide. For example, if a stat account uses "Square Feet

(SQFT)", include that unit abbreviation in the name or description. This reduces confusion for end users and auditors (Source: docs.oracle.com)(Source: blog.concentrus.com).

- **Consistent Units:** Double-check that the saved searches or imports driving each stat account use the correct unit type. NetSuite enforces that all journal lines for a stat account share its unit, so a mismatch causes errors (Source: docs.oracle.com). Best practice is to have a dedicated saved search (with appropriate summary aggregation) for each stat account metric. Test the search first to ensure the numeric values make sense before scheduling.

- **Use Segmentation Wisely:** Only apply Department/Class/Location restrictions if needed. If a stat account is global (all departments), do not restrict it; conversely, if it's department-specific, enforce that restriction on the account (Source: docs.oracle.com). Remember: if you limit a stat account to, say, Class="West", you must include Class=West on every journal entry or schedule line. In OneWorld, plan your subsidiary setup: stat account balances from multiple subs are automatically combined at a 1:1 exchange rate (Source: docs.oracle.com). Use **Include Children** judiciously to cover entire divisions.

- **Schedule Testing and Versioning:** When creating a Statistical Schedule, always **test** it (use the "Create Journal Entry" button in View mode) and review the generated journal (Source: sikich.com) before relying on its output. Keep a log of schedule definitions (which saved search was used, frequency, last run date). If your saved search logic changes (say you add a filter), update or create a new schedule accordingly.

- **Regular Reconciliation:** Periodically reconcile stat account balances with the source data. For example, if Headcount is driven by an HR search, compare the stat account register balance to actual headcount reports. If tracking square footage, ensure each monthly entry matches the operations report. Any discrepancy should trigger an investigation (perhaps a missed journal entry).

- **Audit Trail:** NetSuite tracks each statistical journal's history and provides a *Statistical Account Register*. Auditors can click a stat account in the Chart of Accounts to see all its entries (Source: docs.oracle.com). Encourage reviewers to use this register – or better yet, build a custom report – to spot anomalies (such as unexpected jumps). Documenting rationale for any manual "adjustment" entries (using the Memo field, which highlights offset journals) helps maintain transparency (Source: docs.oracle.com).

- **Unit Type Governance:** Since unit types are global configuration records, avoid changing or deleting a unit once it's in use. The base unit for a type cannot be changed after a stat account uses it (Source: docs.oracle.com). If you need a new unit (e.g. from SQFT to SQM), create a

new unit type and stat account, rather than repurposing the old one.

- **Version Control for Absolute Updates:** If using absolute-balance journals, remember NetSuite uses the **system date** on offset journals (Source: docs.oracle.com). To maintain historical accuracy, ensure offset journals are entered on the correct dates (backdated if needed). Analysts should be aware of whether a stat account holds "delta" entries or absolute snapshots so they query it correctly (by summing or by taking the latest value) (Source: reddit.com) (Source: docs.oracle.com).

These practices help ensure that statistical accounts remain accurate over time and that their usage is clear to any reviewer. When implemented carefully, stat accounts can greatly enhance reporting without compromising data integrity.

Common Pitfalls and How to Avoid Them

Working with statistical accounts has some "gotchas" to watch out for:

- **Feature Not Enabled:** A very common mistake is forgetting to enable the Statistical Accounts feature. Without it, the system won't allow creating stat accounts or schedules (Source: technologyblog.rsmus.com). Similarly, if you plan to use dynamic allocations, make sure **Dynamic Allocation** is enabled; otherwise your stat account cannot be added to an allocation schedule (Source: docs.oracle.com).

- **Misaligned Unit Type:** All lines in a stat journal must use the account's unit of measure. A mismatch (e.g. counting in feet when the unit is set to square meters) triggers an error (Source: docs.oracle.com). Always verify that your saved search (or journal import) returns values in the correct unit.

- **Segmentation Mismatch:** If you restrict a stat account by Department/Class/Location, each journal entry must include those segments. Forgetting to do so causes the entry to be invalid. Moreover, as noted in the setup, "the segments you define in the statistical account are the only segments that can be used in the allocation schedule" (Source: docs.oracle.com). For example, if you want to allocate by Subsidiary and Department, the stat account must also include both Subsidiary and Department segments. Mismatched segments will block the schedule from running properly.

- **Overwriting Data Unintentionally:** Using absolute-balance journals can overwrite history if done incorrectly. For instance, entering an absolute value for the wrong period (or date) might wipe out the previous balance. Always double-check the memo on an offset journal ("Absolute

Balance Update") to ensure it's intended, and consider locking or restricting who can post absolute updates.

- **Currency Confusion (OneWorld):** Remember stat accounts have no currency conversion. If you assign a stat account to multiple subsidiaries, NetSuite *combines* all values at a 1:1 exchange rate (Source: docs.oracle.com). This means a stat account in USD sub.1 and a EUR sub.2 will simply add their balances together (ignoring currency), which may or may not match intuitive expectations. If you need separate tracking by currency, use separate stat accounts.

- **Saved Search Errors:** When scheduling, a too-broad saved search can generate unexpected results. For example, if your employee search accidentally includes terminated staff, headcount will be inflated. Always include necessary criteria/filters. Also note that running the saved search manually does **not** post anything – you must run the statistical schedule (Source: docs.oracle.com). A best practice is to "Validate Saved Search" in the schedule setup to preview results before saving (Source: sikich.com).

- **Ignoring Register Reviews:** Failing to review the Statistical Account Register is a pitfall. Unlike normal journals, stat journals might be overlooked in a periodic close. Ensure someone reviews the stat account entries to confirm they match source data.

By being aware of these pitfalls and using the tips above, you can avoid errors in your NetSuite statistical accounting.

Conclusion

NetSuite Statistical Accounts are a powerful feature for linking operations data to financial analysis. When properly configured and maintained, they allow organizations to include any non-financial metric in budgets, allocations, and KPI reporting. This capability is invaluable for advanced costing (e.g. Activity-Based Costing), performance measurement (e.g. EPS, customer metrics), and strategic decision-making (Source: docs.oracle.com)(Source: netsuite.folio3.com). As always, success depends on following best practices: enable the feature, define clear unit types, maintain accurate schedules, and audit the balances regularly. With these elements in place, financial controllers and NetSuite administrators can wield statistical accounts to gain deeper insights and drive data-informed decisions across all industries.

Sources: Official NetSuite documentation and implementation guides (Source: docs.oracle.com) (Source: docs.oracle.com) (Source: docs.oracle.com)(Source: docs.oracle.com) (Source: technologyblog.rsmus.com); NetSuite consultant blogs and tech articles (Source:

netsuite.folio3.com)(Source: houseblend.io) (Source: sikich.com)(Source: netsuite.folio3.com) (Source: houseblend.io) (cited above). All information is current as of 2025.

Tags: netsuite, statistical accounts, financial accounting, general ledger, non-financial metrics, reporting, budgeting, cost allocation, erp, chart of accounts

About Houseblend

HouseBlend.io is a specialist NetSuite™ consultancy built for organizations that want ERP and integration projects to accelerate growth—not slow it down. Founded in Montréal in 2019, the firm has become a trusted partner for venture-backed scale-ups and global mid-market enterprises that rely on mission-critical data flows across commerce, finance and operations. HouseBlend's mandate is simple: blend proven business process design with deep technical execution so that clients unlock the full potential of NetSuite while maintaining the agility that first made them successful.

Much of that momentum comes from founder and Managing Partner **Nicolas Bean**, a former Olympic-level athlete and 15-year NetSuite veteran. Bean holds a bachelor's degree in Industrial Engineering from École Polytechnique de Montréal and is triple-certified as a NetSuite ERP Consultant, Administrator and SuiteAnalytics User. His résumé includes four end-to-end corporate turnarounds—two of them M&A exits—giving him a rare ability to translate boardroom strategy into line-of-business realities. Clients frequently cite his direct, "coach-style" leadership for keeping programs on time, on budget and firmly aligned to ROI.

End-to-end NetSuite delivery. HouseBlend's core practice covers the full ERP life-cycle: readiness assessments, Solution Design Documents, agile implementation sprints, remediation of legacy customisations, data migration, user training and post-go-live hyper-care. Integration work is conducted by in-house developers certified on SuiteScript, SuiteTalk and RESTlets, ensuring that Shopify, Amazon, Salesforce, HubSpot and more than 100 other SaaS endpoints exchange data with NetSuite in real time. The goal is a single source of truth that collapses manual reconciliation and unlocks enterprise-wide analytics.

Managed Application Services (MAS). Once live, clients can outsource day-to-day NetSuite and Celigo® administration to HouseBlend's MAS pod. The service delivers proactive monitoring, release-cycle regression testing, dashboard and report tuning, and 24 × 5 functional support—at a predictable monthly rate. By combining fractional architects with on-demand developers, MAS gives CFOs a scalable alternative to hiring an internal team, while guaranteeing that new NetSuite features (e.g., OAuth 2.0, AI-driven insights) are adopted securely and on schedule.

Vertical focus on digital-first brands. Although HouseBlend is platform-agnostic, the firm has carved out a reputation among e-commerce operators who run omnichannel storefronts on Shopify, BigCommerce or Amazon FBA. For these clients, the team frequently layers Celigo's iPaaS connectors onto NetSuite to automate fulfilment, 3PL inventory sync and revenue recognition—removing the swivel-chair work that

throttles scale. An in-house R&D group also publishes “blend recipes” via the company blog, sharing optimisation playbooks and KPIs that cut time-to-value for repeatable use-cases.

Methodology and culture. Projects follow a “many touch-points, zero surprises” cadence: weekly executive stand-ups, sprint demos every ten business days, and a living RAID log that keeps risk, assumptions, issues and dependencies transparent to all stakeholders. Internally, consultants pursue ongoing certification tracks and pair with senior architects in a deliberate mentorship model that sustains institutional knowledge. The result is a delivery organisation that can flex from tactical quick-wins to multi-year transformation roadmaps without compromising quality.

Why it matters. In a market where ERP initiatives have historically been synonymous with cost overruns, HouseBlend is reframing NetSuite as a growth asset. Whether preparing a VC-backed retailer for its next funding round or rationalising processes after acquisition, the firm delivers the technical depth, operational discipline and business empathy required to make complex integrations invisible—and powerful—for the people who depend on them every day.
