Billing Rules

Billing rules tell `TMSS10LogStatsProcess` how to convert Customer activity from a `.ttstats` file into invoice items.

Rules are stored in a billing definition file, normally with the `.ttsdef` extension. Each rule filters a set of operational records, calculates a billable quantity, and writes an XML `<item>` for QuickBooks.



Always run `TMSS10LogStatsProcess -l` and compare the displayed signatures with the rules before deploying a new definition. The installed utility is the authoritative source for the version in use.

How Rules Are Selected

A rule is evaluated only when its filters match the current Customer's data.

The available filters depend on the rule type, but may include:

- Billing package.
- Customer-ID.
- Media-ID.
- Repository-ID.
- Source and destination Repository-ID.
- Journal category.

The package is normally assigned in a package map supplied with the processor's `-P` option:

packagemap.ttmap

```
US01 US07_CUSTOM
US02 Tape_Daily
US03 Box_Weekly
```

When `US01.ttstats` is processed, `US01_CUSTOM` becomes the package value. Rules filtered to `US01_CUSTOM` can match, while rules for `Tape_Daily` or `Box_Weekly` do not.

Supported Rule Types

Rule	Data used	Typical purpose
`AddStorageChargeItem`	Daily storage and slot records	Tape storage, repository storage, base fees, and transport-container use.

Rule	Data used	Typical purpose
AddInventoryChargeItem	Daily inventory records	Container inventory, owned boxes, scanners, and other inventory-based charges.
AddMovementChargeItem	Daily movement records	Deliveries, pickups, handling, and movements above a package allowance.
AddJournalChargeItem	Journal category and quantity records	Scheduled trips, urgent service, labour, reports, materials, and other recorded services.
AddAppendFile	External text or XML file	Fixed, recurring, annual, quarterly, or contract-specific invoice content.

Filter Syntax

The supplied definitions demonstrate these common filter patterns:

Filter	Meaning
“*”	Match any value.
“US01”	Match the exact value US01.
“Tape*”	Match values beginning with Tape.
“B*”	Match values beginning with B.
“*(LT|DL|LTO)”	Match one of the grouped Media-ID values.
“*(COV|RVLT|OFFS)”	Match one of the grouped Repository-ID values.
“!(HOLD)”	Match any value except HOLD.
“!(US02|US03)”	Match any value except the listed values.

Broad filters can include activity that was not intended to be billed. For example, a movement rule with source !(HOLD) and destination * also matches OFFS to OFFS internal movements.

Common Calculation Arguments

After matching operational records, the processor applies the calculation arguments in this order:

1. Round up.
2. Enforce the minimum.
3. Apply the threshold.
4. Apply the cap.

Argument	Purpose	Example
Round Up Factor	Rounds a non-zero quantity to the next multiple.	50 converts 1–50 to 50 and 51–100 to 100.
Minimum Units	Raises a non-zero quantity to the contractual minimum.	A calculated quantity of 12 with minimum 25 becomes 25.
Threshold	Represents included units. The threshold is subtracted when the quantity exceeds it; otherwise the result is zero.	250 movements with threshold 230 produces 20 billable movements.

Argument	Purpose	Example
Cap	Limits the final quantity. A value of zero means no cap.	A cap of 1 is commonly used to produce a single base-fee item.
Model	Selects the aggregate used by the rule.	Storage model 1 selects real high-water.

If the final quantity is zero, the processor does not create an invoice item.

Calculation Example

A movement package includes 230 movements. During the billing period the Customer has 275 qualifying movements:

- Calculated total: 275.
- Round Up Factor: 0, so the total remains 275.
- Minimum Units: 0, so the total remains 275.
- Threshold: 230, leaving 45.
- Cap: 0, so no upper limit is applied.
- Invoice quantity: 45.

Models

Storage Models

Model	Meaning
0	Real total: sum of daily storage counts. This represents tape-days for a daily billing period.
1	Real high-water: highest daily storage count.
2	Real low-water: lowest daily storage count.
3	Slot total.
4	Slot high-water.
5	Slot low-water.

Movement, Inventory, and Journal Models

Model	Movement	Inventory	Journal
0	Total movements	Total inventory	Total journal quantity
1	Movement high-water	Inventory high-water	Journal high-water
2	Movement low-water	Inventory low-water	Journal low-water

Storage Rules

The observed storage rule structure is:

```
AddStorageChargeItem(
  packageFilter,
  customerFilter,
  mediaFilter,
```

```
  repositoryFilter,
  roundUpFactor,
  minimumUnits,
  threshold,
  cap,
  model,
  outputCustomer,
  outputItemCode,
  description,
  costInCents,
  continueFlag
);
```

The following rule bills US01 LTO storage in OFFS at 2 cents per tape-day:

```
AddStorageChargeItem("US01_CUSTOM", "US01", "LTO", "OFFS",
                     0, 0, 0, 0, 0,
                     "$CUSTOMER2", "US01_LTO_STORAGE",
                     "LTO tape storage at New York OFFS",
                     2, 0);
```

Position	Value	Explanation
Package	US01_CUSTOM	Only the package assigned to US01 can match.
Customer	US01	Prevents another Customer in the package from matching.
Media	LTO	Only LTO records are used.
Repository	OFFS	Only storage in the offsite repository is billed.
Round, minimum, threshold, cap	0, 0, 0, 0	No quantity adjustments.
Model	0	Sums daily counts to produce tape-days.
Output Customer	$CUSTOMER2	Uses the QuickBooks Customer name from the Customer map.
Item Code	US01_LTO_STORAGE	QuickBooks Item reference.
Cost	2	Two cents per unit.
Continue	0	Matching storage data is spent and cannot be billed again by a later storage rule.

If the daily counts are 100, 105, and 110 tapes, model 0 produces 315 tape-days. At 2 cents per tape-day, the calculated charge is $6.30.

Movement Rules

The observed movement rule structure is:

```
AddMovementChargeItem(
  packageFilter,
  customerFilter,
  mediaFilter,
```

```
  fromRepositoryFilter,
  toRepositoryFilter,
  roundUpFactor,
  minimumUnits,
  threshold,
  cap,
  model,
  outputCustomer,
  outputItemCode,
  description,
  costInCents,
  continueFlag
);
```

Delivery

```
AddMovementChargeItem("US01_CUSTOM", "US01", "LTO", "OFFS", "CUST",
                      0, 0, 0, 0, 0,
                      "$CUSTOMER2", "US01_LTO_DELIVERY",
                      "LTO tape delivery from OFFS to CUST",
                      5, 0);
```

This rule counts LTO tapes moved from `OFFS` to `CUST`. Model `0` totals the matching movements and each tape is charged at 5 cents.

Pickup

```
AddMovementChargeItem("US01_CUSTOM", "US01", "LTO", "CUST", "OFFS",
                      0, 0, 0, 0, 0,
                      "$CUSTOMER2", "US01_LTO_PICKUP",
                      "LTO tape pickup from CUST to OFFS",
                      3, 0);
```

This rule counts LTO tapes moved from `CUST` to `OFFS`. Each matching tape is charged at 3 cents.

Free Internal Movements

No rule is defined for:

```
OFFS -> OFFS
```

Because neither the delivery nor pickup rule matches that route, an internal movement does not create an invoice item.



Do not create a zero-cost `OFFS` to `OFFS` rule. A cost of `0.00` causes the QuickBooks



Interface to omit the Rate field, which may allow the QuickBooks Item's default rate to be applied. Leave free activity unmatched and ensure that no later wildcard rule can match it.

Inventory Rules

An inventory rule filters by package, Customer, and Media-ID. It does not include repository filters in the observed definition format.

```
AddInventoryChargeItem(
  packageFilter,
  customerFilter,
  mediaFilter,
  roundUpFactor,
  minimumUnits,
  threshold,
  cap,
  model,
  outputCustomer,
  outputItemCode,
  description,
  costInCents,
  continueFlag
);
```

Example:

```
AddInventoryChargeItem("No_Package", "*", "B1",
                       0, 0, 0, 0, 1,
                       "$CUSTOMER", "A - Storage:QBSTORAGE_B1",
                       " ", 0, 1);
```

This example selects high-water inventory for Media-ID B1. A zero cost means that QuickBooks Item pricing is expected to supply the rate.

Journal Rules

Journal rules convert a journal category and recorded quantity into an invoice item.

```
AddJournalChargeItem(
  packageFilter,
  customerFilter,
  journalCategoryFilter,
  roundUpFactor,
  minimumUnits,
```

```
  threshold,
  cap,
  model,
  outputCustomer,
  outputItemCode,
  description,
  costInCents,
  continueFlag
);
```

Example:

```
AddJournalChargeItem("Tape_Daily*", "*", "00001",
                     0, 0, 23, 0, 0,
                     "$CUSTOMER",
                     "B - General Services:QBPICKUP_SCHEDULED_NOT_INCLUDED",
                     " ", 0, 1);
```

This rule applies to daily tape packages and journal category `00001`. The first 23 qualifying units are included; only the quantity above 23 is invoiced.

The free-text note stored in a journal record is not used by the documented billing calculation. Rules select the category and quantity.

Append Files

`AddAppendFile` loads an external file and adds its content to the invoice output after variable substitution.

```
AddAppendFile("No_Package", "US01", "..\etc\billing\US01_Monthly.txt");
```

Append files may be used for fixed fees, recurring charges, credits, or Customer-specific contract items.

Append-file content is written directly into the processor output. Keep every append file under version control and validate the resulting XML before import.

Output Variables

Variable	Source	Typical use
`$CUSTOMER`	The C record in the `.ttstats` file	TapeTrack Customer-ID.
`$CUSTOMER2`	Customer description map supplied with `-M`	QuickBooks Customer FullName or another mapped description.

Customer map example:

customer.ttmap

```
US01 New York Data Center
```

For this Customer, `$CUSTOMER` becomes `US01` and `$CUSTOMER2` becomes `New York Data Center`.

Pricing

The cost argument is expressed in cents:

Rule value	Invoice rate
2	$0.02
3	$0.03
5	$0.05
100	$1.00

The processor multiplies the final quantity by the cost override when producing its XML values.

When cost is zero, the QuickBooks Interface omits the `Rate` element. QuickBooks may then use the default rate configured for the referenced Item.

Do not assume that a zero cost makes an item free. Use no matching rule for free activity, or verify the intended pricing behavior in a QuickBooks test company.

A calculated XML cost of `99999.99` is treated by the documented QuickBooks Interface as a signal to skip the line. This should not be used as a normal price or as the standard method for excluding activity.

Continue and Rule Order

The Continue flag controls whether matched aggregate data remains available to later rules of the same type:

Value	Behavior
0	Stop for the matched data. The data is marked as spent and cannot be billed again by a later rule of the same class.
Non-zero	Continue. The matched data remains available to later rules.

Rule order therefore matters. Place Customer exceptions before broader package or wildcard rules when the exception must consume the data.

Example of an unsafe sequence:

```
# Specific free rule, but Continue is enabled
AddMovementChargeItem("*", "US01", "LTO", "OFFS", "OFFS",
                      0, 0, 0, 0, 0,
                      "$CUSTOMER2", "INTERNAL_MOVE", " ", 0, 1);

# Broad rule can still match the same movement
AddMovementChargeItem("*", "*", "LTO", "!(HOLD)", "*",
                      0, 0, 0, 0, 0,
                      "$CUSTOMER", "QBMOVEMENT_TAPE", " ", 0, 1);
```

The first rule does not reliably make the movement free. It produces a zero-rate item and leaves the movement available to the broad rule.

The safer design is to omit the internal-movement rule and ensure that the broad rule excludes the Customer or route.

Processing Order

The processor calculates rule classes in this order:

1. Storage.
2. Inventory.
3. Movement.
4. Journal.
5. Append files.

Within each class, rules are applied in definition order.

Validation Checklist

Before approving a definition:

- Run `TMSS10LogStatsProcess -l` and confirm every signature.
- Confirm the package map assigns the Customer to the expected package.
- Confirm Customer, Media-ID, and Repository-ID values match TapeTrack exactly.
- Check source and destination direction on every movement rule.
- Search for wildcard movement rules that could match internal activity.
- Confirm calculation order for rounding, minimum, threshold, and cap.
- Confirm the selected model represents the contract.
- Confirm whether pricing comes from the rule or from QuickBooks.
- Review Continue values and rule order for duplicate billing.
- Generate a test XML file from a known `.ttstats` sample.
- Confirm every Customer and Item reference exists in QuickBooks.
- Recalculate representative invoice quantities independently.
- Confirm free activities produce no XML item.
- Import into a QuickBooks test company before production use.

Common Problems

Symptom	Likely cause
No invoice item is produced	A package or data filter did not match, or calculation adjustments reduced the quantity to zero.
Wrong quantity	Incorrect model, an unexpected threshold, wrong extraction options, or broader filters than intended.
Duplicate invoice lines	Overlapping rules combined with Continue enabled.
Internal moves are charged	A wildcard movement rule matches both the source and destination.
Zero-cost item has a price in QuickBooks	The interface omitted Rate and QuickBooks applied the Item's default price.
Wrong QuickBooks Customer	Incorrect `$CUSTOMER2` map value or Customer reference mode.
Expected package rules do not run	Missing or incorrect package-map entry.
A rule never matches	Typographical error, inconsistent capitalisation, or invalid grouped-filter syntax.

Related Pages

- Implementing Customer Billing
- Adding and Maintaining Customers
- Customer Object

From:
https://docs.tapetrack.com/ - **TapeTrack Documentation**

Permanent link:
https://docs.tapetrack.com/billing/billing_rules?rev=1787105527

Last update: **2026/08/19 02:12**