

JCC Stockroom Unit Conversion System Upgrade Documentation

****Date:**** 2025-05-27

****Author:**** Nova

****Scope:**** Conversion logic upgrade, UI restrictions, and module integration for `item-info.js`, `waste-entry.js`, and `dataUtils.js`.

Overview of the Upgrade

Goal:

Transform the unit conversion system into a ****smart, flexible, and context-aware engine**** that:

- Standardizes all conversions using the ****Case Unit Type**** as a base.
- Supports ****multi-hop and bi-directional conversions****.
- Dynamically filters unit selection dropdowns based on compatibility.
- Uses `unitPerCase` to compute ****CS EA**** conversions accurately.
- Locks `Case Unit Type` after an item is saved to prevent downstream data conflicts.

`dataUtils.js` Conversion Engine Overhaul

Key Changes:

- Added ****unit categories****: `weight`, `volume`, `each`
- Created a ****bi-directional, multi-hop conversion system**** using BFS (Breadth-First Search)
- Added ****dynamic CS EA logic**** using `item.unitPerCase`
- New function: `getConvertibleUnits(baseUnit)` returns compatible units for dropdown filtering

New Exports:

```js

```
convertUnits(from, to, quantity, item?) // Multi-hop, supports CS EA
convertWasteQuantity(quantity, fromUnit, item) // Converts & estimates cost
getConvertibleUnits(baseUnit) // Returns compatible UOMs
```

```

dataUtils-upgraded.js

`item-info.js` Locking Case Unit After Save

Key Changes:

- When editing an existing item, the `Case Unit Type` dropdown becomes ****disabled****.
- When adding a new item, the dropdown remains editable.

Code Behavior:

```
```js
if (currentItem?.id) {
 caseUnitField.disabled = true;
} else {
 caseUnitField.disabled = false;
}
```
```

item-info-patched.js

`waste-entry.js` Dynamic Unit Filtering

Key Changes:

- Imports `convertUnits` and `getConvertibleUnits` from `dataUtils.js`
- Adds a function to populate the unit dropdown with compatible options:

```
```js
function updateUnitDropdown(baseUnit) {
 const validUnits = getConvertibleUnits(baseUnit);
 ...
}
```
```

waste-entry-patched.js

How It Works in Practice

1. **Case Unit Becomes Base**

- When an item is created, its `Case Unit Type` is saved.
- After saving, it becomes the **locked base** for all future conversions.

2. **Dropdown Filtering**

- Only compatible units appear in:
 - Default Waste UOM in `item-info.js`
 - Unit field in `waste-entry.js`

3. **Smart Conversions**

- `convertUnits("OZ", "LB", 32)` → `2`
- `convertUnits("FL OZ", "GAL", 64)` → `0.5`
- `convertUnits("EA", "CS", 12, item)` depends on `item.unitPerCase`

Files That Must Be Updated in Production

| File | Purpose |
|--------------------------------|---------------------------------------|
| scripts/utils/dataUtils.js | Core conversion logic |
| scripts/modules/item-info.js | Locks base unit field post-save |
| scripts/modules/waste-entry.js | Smart unit dropdown + conversion prep |

Next Steps / Optional Enhancements

- 1. **Integrate `updateUnitDropdown()` calls in HTML forms**
- 2. **Add client-side validation to prevent submitting invalid conversions**
- 3. **Log unit mismatches or bad conversions for debugging**
- 4. **Apply this logic to future modules (Events, Menus, Reporting)**