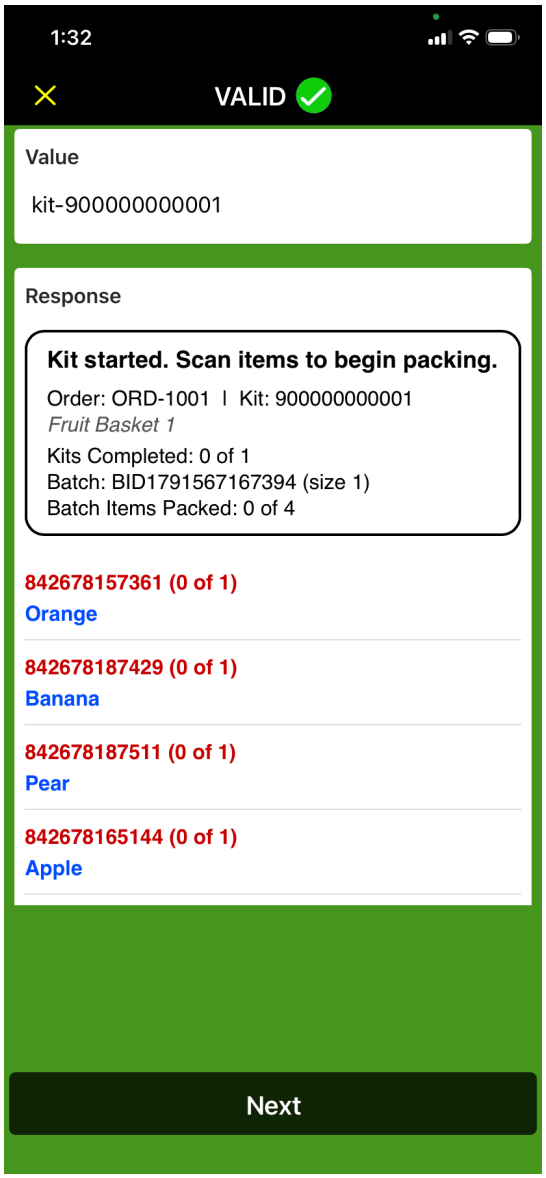
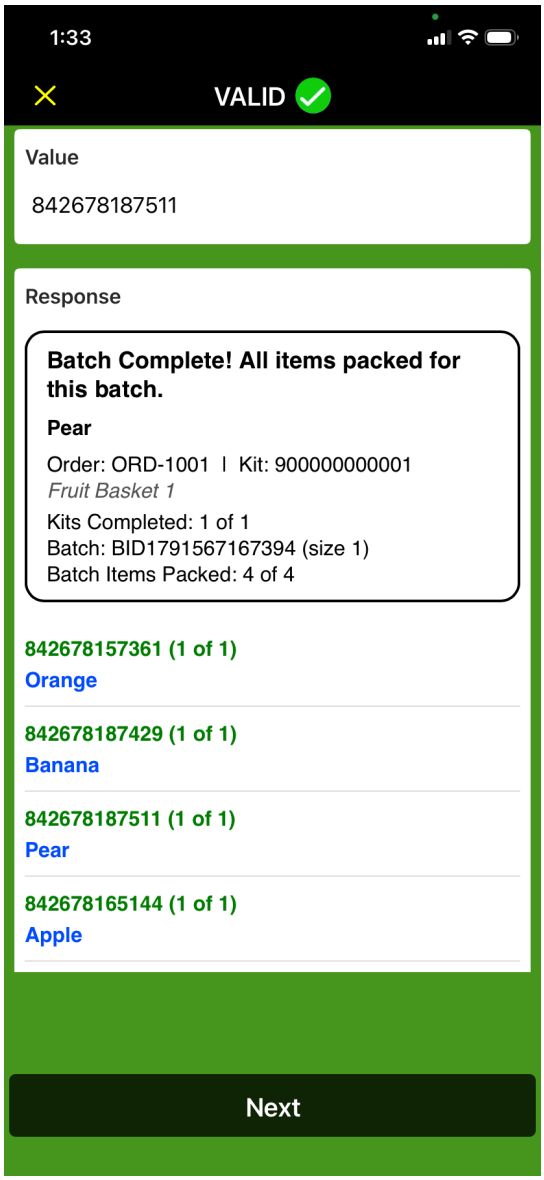
Kitting COV Script — How to Use It

Updated Sep 22, 2026





Overview

This Custom On-Device Validation (COV) script powers a kitting workflow where operators pack multiple items together into "kits" and track progress toward completing an order. Note: This version assumes the same device will be used to start and complete a kit. Ask our support team for alternatives if that restriction won’t work for your use case.

Service setup required:

- Service type: **Validate Scans On-Device**
- Duplicate scans: **Valid** (set at the service level)

Duplicate scans must stay Valid at the service level because a single kit can legitimately require more than one of the same item (e.g., 5 of the same bracelet). All kit-level duplicate/restart logic is handled inside the script itself, not by the platform's built-in duplicate check.

IMPORTANT: This service needs a Custom On-Device (COV) script added on the service's Advanced Step form field after checking "Enable on-device custom validation". Copy and paste this script in the form.

Two kinds of scans

The script recognizes exactly two categories of scanned values, distinguished automatically:

1. **Kit ID scans** — any scanned value starting with the prefix `kit-` (e.g., `kit-0001`). If that value exists in the associated database, it is treated as the start of a new kit count. Scanning a `kit-` value always starts a fresh kit, replacing whatever kit was previously in progress — no reset barcode is needed, even for a kit that's already complete or partially packed.
2. **Item scans** — any other scanned value. These are checked against the currently active kit's item list and counted toward that kit's required quantities.

There is no separate "session question" step for Kit ID, Order ID, Total Qty, or Batch Qty — all of that information lives in the Kit ID's own database record and loads automatically the moment that Kit ID is scanned.

Database Setup

Kit ID records

Each Kit ID (any value starting with `kit-`) needs its own database record whose response is a JSON object with four fields:

```
{
  "orderId": "ORD-1006",
  "totalQty": 4,
  "batchQty": 2,
  "kit": {
    "842678187429": 1,
    "842678187511": 1
  }
}
```

Field	Meaning
`orderId`	The work order this kit belongs to (display only — doesn't affect counting)

Field	Meaning
`totalQty`	Total number of finished kits needed for this order
`batchQty`	How many of those kits are packed together as one batch
`kit`	The recipe for one kit: item UPC → quantity needed per kit

Item records

Each item UPC referenced inside a kit's `kit` object also needs its own separate database record, whose response is simply the item's description text (e.g., `Hope Classic Bracelet - Sherbert`). This is looked up automatically both when a kit starts (to build the item list shown on screen) and every time a matching item is scanned (to show which item was just added).

Scanning Workflow

1. **Scan the Kit ID barcode** (e.g., `kit-0006`). The app looks it up in the database, loads its order info and item recipe, and starts a fresh count at 0 for however many kits the order needs. The response screen shows the order, kit ID, kits completed, current batch, and the full item list with quantities needed.
2. **Scan item barcodes** to pack the kit. Each valid item increments its count toward that batch's requirement. Items may be scanned in any order, and the same item can be scanned multiple times if the kit needs more than one.
3. **When a batch's items are fully packed**, the screen shows "Batch Complete!" and the next scan automatically starts a new batch (if more kits remain in the order).
4. **When all kits in the order are packed**, the screen shows "Order Complete!" instead.

Rescanning a Kit ID

Scanning any `kit-` value — the same one currently in progress, a different one, complete or not — immediately abandons whatever was in progress and starts that kit fresh from zero. No reset barcode is required for this. This means operators should be careful: accidentally rescanning a Kit ID barcode (e.g., a double-fired scanner) will silently discard in-progress counts for the kit that was active.

Understanding totalQty and batchQty

`totalQty` is how many finished kits the whole order needs. `batchQty` is how many of those kits get packed together as one batch before the app gives a checkpoint message.

Worked example: `kit-0006`

```
{"orderId": "ORD-1006", "totalQty": 4, "batchQty": 2, "kit": {"842678187429": 1, "842678187511": 1}}
```

This kit needs 1 of each item per kit, 4 kits total, packed 2 kits at a time (2 batches overall).

Step	What happens
Scan `kit-0006`	Kits completed: 0 of 4. This batch needs 2× each item (2 of each barcode)
Scan each item twice	Kits completed: 2 of 4. "Batch Complete!"
Scan next item	Counts reset for a new batch, still needing 2× each item
Scan each item twice again	Kits completed: 4 of 4. "Batch Complete!" and "Order Complete!" fire together

Single-batch case: when `totalQty` equals `batchQty` (e.g., both 5, as in `kit-0010`), there's only one batch — no intermediate "Batch Complete" checkpoint, just "Order Complete" once all 5 kits' worth of items are scanned.

Reset Barcodes

Two special barcodes remain for clearing state without scanning a new Kit ID:

Barcode value	Effect
`o123` (reset order)	Clears all active kit/order state on this device entirely
`b123` (reset batch)	Clears only the current batch's packed counts, if a batch is in progress; otherwise reports there's nothing to reset

Since rescanning a Kit ID now always starts that kit fresh on its own, these reset codes are mainly useful for clearing state without immediately starting a new kit — for example, to leave the device idle between orders.

Message Reference

Message	Meaning
Kit started. Scan items to begin packing.	A Kit ID scan succeeded; ready for item scans
Item added.	An item scan counted toward the current batch
Batch Complete! All items packed for this batch.	The current batch's item quotas are fully met
Order Complete! All kits packed for this order. Scan a new Kit ID to start another kit.	The whole order's kits are fully packed
This kit's order is already complete. Scan a new Kit ID to start another kit.	An item was scanned after the active order already finished

Message	Meaning
Required quantity for this item is already completed.	That item's quota for the current batch is already met (marked Invalid so Auto-Next Scan pauses)
Scanned item is not part of this kit.	The scanned barcode isn't in the active kit's item list (marked Invalid)
Scanned value is not a recognized Kit ID.	A `kit-` prefixed value was scanned but has no matching database record
Scan a Kit ID barcode to begin.	An item was scanned with no active kit in progress
There is an issue with this Kit ID's database record.	The Kit ID's database record is missing required fields (order ID, quantities, or item list)
You can start a new work order now.	The reset-order barcode was scanned
You can start a new batch now.	The reset-batch barcode was scanned while a batch was in progress
There is no batch in progress to reset.	The reset-batch barcode was scanned with no batch active