

The POS Switch Playbook

Migrating from your old point of sale without losing a service

What this is: a six-week operational playbook for replacing a restaurant point-of-sale system. It covers the data you must extract before you cancel, the menu and modifier rebuild, tax and tender configuration, printer and kitchen screen routing, the payments cutover and first-week reconciliation, staff training, an hour-by-hour cutover runbook, and how to cancel the old contract without paying for a year you did not want. **Who it is for:** owners and general managers running the switch themselves, and anyone about to sign with a new vendor who wants to know what to ask first.

Free operator resource · Version 1.0 · Updated August 2026

Practical guidance only. Contract, tax and payment rules vary by jurisdiction and by agreement - read your own contract and confirm specifics with your lawyer and accountant.

Why operators switch - and when not to

Changing point of sale is one of the few projects that touches every part of a restaurant at once: the menu, the kitchen, the floor, payroll, payments and the books. Done well it is a boring weekend. Done badly it costs you a Saturday night, a month of clean reporting, and the trust of a team who now believe the new system is the problem.

This playbook assumes you have already decided to move. It exists to make sure the move does not cost you a service.

Reasons that genuinely justify a switch

- **Total cost is out of line with what you get.** Not the headline software price - the all-in number including per-terminal fees, module fees, payment processing and hardware. Compare effective rates on your real card mix, not advertised rates.
- **You cannot get your own data out.** If you cannot export menu, sales history, guests and loyalty balances yourself, you do not control your business.
- **The system cannot model your menu.** Modifier limitations that force staff into workarounds show up later as wrong reporting and wrong food cost.
- **It fails when the internet fails.** If a network outage stops you taking orders or cards, that is a revenue risk, not an inconvenience.
- **Support does not answer.** Test this during evaluation by calling the support line at 8 pm on a Saturday.
- **Growth is blocked.** A second location, online ordering, catering or delivery integration that the current platform cannot support, or charges unreasonably for.
- **End of life or forced migration.** The vendor is sunsetting your version or your hardware, and you would rather choose than be moved.

Reasons that do not justify a switch

- **Staff dislike the interface.** Nine times out of ten the underlying problem is a badly built menu and no training. Rebuilding your button layout is a one-week project; switching systems is a six-week one.
- **Reports are wrong.** Check first whether categories, modifiers and comp reasons were ever configured properly. A new system inherits your bad category tree if you rebuild it the same way.
- **One bad support experience.** Escalate, get a named account contact, and see whether it repeats.
- **A better price from a competitor.** Take that quote to your incumbent first. Retention pricing is real, and renegotiating costs you no downtime.

DO NOT SWITCH DURING THESE WINDOWS

Never cut over inside your two busiest weeks of the year, during a holiday period, in the week a key manager is on leave, or while you are also opening a patio, launching a new menu or onboarding a delivery channel. Pick your slowest week, and cut over on the slowest night of that week. If your slow night is Monday, cut over Sunday night after close.

The six-week switch timeline

Six weeks is a realistic minimum for a single-location full-service restaurant with a normal menu. Compress it only if you have run a switch before. Multi-site operators should run the full six weeks on the first location and then treat it as a template.

Week	Focus	Key deliverables	Who owns it
Week 6	Decide and contract	Vendor selected; contract and payment processing signed; hardware ordered; kick-off call held; project owner named; cutover date chosen and protected in the schedule.	Owner / GM
Week 5	Extract and audit	Full data export from the incumbent system taken and stored in two places; gift card and loyalty liability reconciled; menu audited and pruned; tax rules documented; old contract notice requirements confirmed in writing.	GM + bookkeeper
Week 4	Build	Categories, items, modifier groups, price levels and taxes built in the new system; floor plan drawn; employee records and permissions created.	GM + vendor
Week 3	Hardware and network	Network drops and power confirmed; terminals, printers, KDS and payment devices installed and staged; routing matrix built and tested with live test tickets; UPS units in place.	Vendor + electrician
Week 2	Test and train	Full menu ring-through test; managers trained first; training sessions run for every role; cheat sheets printed; parallel-run decision made.	GM + trainers
Week 1	Rehearse	Dress rehearsal service on the new system; punch list cleared; cutover runbook printed and assigned; rollback plan agreed; old system data export repeated and verified.	GM
Cutover	Go live	Old system closed out; new system live; first-day reconciliation completed against bank deposit.	Owner + GM + vendor
Week +1	Stabilise	Daily reconciliation; report validation; menu and button fixes; staff feedback loop; old contract cancelled only after data is verified.	GM

OPERATOR TIP

Name one person as the project owner and put their name on the plan. Not "management", not "the vendor" - a person. Switches fail in the seams between the vendor's assumption that you are building the menu and your assumption that they are.

Data you must extract before you cancel

This is the single most important section of this playbook. The moment your old account is cancelled or lapses for non-payment, your access to historical data usually ends - sometimes immediately, sometimes after a short grace window you will not be told about. Extract everything **while the account is fully active and paid**, verify the exports open and contain what you expect, and store two copies in two different places.

What to extract	Format to insist on	Why you will need it	Verify by
Full menu: categories, items, prices, descriptions, SKUs or PLUs	CSV or Excel, one row per item	The base of the rebuild, and your evidence if the new build drifts from the old prices.	Row count matches your item count; spot-check twenty prices against the printed menu.
Modifier groups and modifiers, with prices and min/max rules	CSV, or screenshots if the system will not export	Modifier logic is where rebuilds go wrong. If you cannot export it, photograph every modifier screen.	Rebuild one complex item from the export alone and see whether it matches.
Price levels, happy hour and time-based pricing	CSV or documented screenshots	Time-based rules rarely export cleanly and are almost always missed.	Write out each rule in plain English: item, days, times, price.
Tax configuration: rates, which items are taxable, dine-in versus take-out treatment, alcohol treatment	Written documentation, plus sample receipts	Getting tax wrong on day one creates a filing problem and a refund problem.	Keep one printed receipt of each tax case as the reference.
Guest and loyalty database: names, contact details, consent flags, points or stored value balances	CSV with balances as at a stated date and time	This is your marketing list and a real financial liability.	Sum the balances and reconcile to the liability on your books.
Gift card balances, card numbers and issue dates	CSV with outstanding balance per card, as at a stated timestamp	Outstanding gift cards are money you owe your guests. See the callout on the next page.	Total outstanding matches the liability account in your bookkeeping.
Sales history: daily sales by category, by day-part, by payment type, at least 24 months	CSV at daily grain, plus item-level if available	Without it you have no year-over-year comparison, no seasonality model and no budget baseline.	Recreate one month's total from the export and match it to that month's financial statements.
Product mix / item sales history	CSV, item by period, units and net sales	Menu engineering, forecasting and par levels all depend on it.	Top ten items by units look like what you expect.
Employee records: names, roles, wage rates, hire dates, PINs or IDs	CSV	Rebuilding staff records by hand invites errors in permissions and payroll.	Count matches your current roster.
Labour history: hours by employee by day, and by job code	CSV	Needed for scheduling, labour benchmarking and any wage or hours dispute.	One week's hours matches one week's payroll register.
Open house accounts, tabs and account receivables	Report plus written list	Anything owed to you that is only recorded in the old POS disappears with it.	Zero the balances before cutover, or transfer them in writing.

What to extract	Format to insist on	Why you will need it	Verify by
Open or future reservations, deposits and catering bookings	CSV plus confirmation emails	Guests will arrive with a booking you cannot see.	Re-enter into the new system and re-confirm with each guest.
Recipes, ingredient costs and inventory counts, if held in the POS	CSV	Recreating recipe costing is weeks of work.	Spot-check five recipes against your recipe cards.
Historical voids, comps, discounts and cash over/short	CSV	Needed if you are ever audited, and useful as a control baseline.	Totals reconcile to the same period's sales reports.
Copies of daily closeout reports for the last full fiscal year	PDF	Auditors and tax authorities ask for these. Your accountant will thank you.	One file per day, no gaps.

GIFT CARD AND LOYALTY LIABILITY - READ THIS TWICE

Outstanding gift card balances are a genuine financial liability of your business. They do not disappear because your POS changed. A guest holding a card you sold last Christmas is entitled to spend it, and in many jurisdictions gift cards for food and beverage cannot legally expire. Before cutover you must: (1) export every outstanding balance with a timestamp; (2) reconcile the total to the gift-card liability on your balance sheet and investigate any gap; (3) decide and document how balances will be honoured - either migrated into the new system as loaded cards, or held on a manual register that every manager can access and that is reconciled weekly; (4) brief every manager and every cashier, in writing, on exactly what to do when an old card is presented; and (5) keep the old register for at least as long as your jurisdiction's rules require. The same logic applies to loyalty points with a monetary value. Never cancel the old account until this is done - once it is closed, the balances are gone and every dispute becomes your word against the guest's.

How to run the extraction

- ☐ Ask the incumbent, in writing, for a complete data export and for their documented data-retention and post-cancellation access policy. Keep the reply.
- ☐ Take the export yourself from the back office as well - do not rely solely on what the vendor sends.
- ☐ Take exports twice: once in week 5, and again in the final 48 hours before cutover so the last days of trading are captured.
- ☐ Screenshot anything that will not export: modifier screens, tax rules, discount configurations, printer routing, permission matrices.
- ☐ Open every file. An export that produces a zero-byte CSV or a report that only shows the last 90 days is a problem you want to find in week 5, not week 1.
- ☐ Store one copy in cloud storage the business owns (not a personal account) and one copy offline. Name files with the date and the source system.
- ☐ Give your bookkeeper and accountant a copy of the sales history and closeout reports before cutover.

Rebuilding the menu

Do not port the old menu button for button. A switch is the one moment when restructuring costs you nothing extra. Rebuild it the way you wish it had been built, then map the old items to the new structure so your history stays comparable.

Categories: build for the report you want to read

- Your category tree determines every sales report you will read for years. Design it backwards from the report, not forwards from the menu layout.
- Keep top-level categories few and stable - typically food, beverage, alcohol by type, retail, other. Put the detail in subcategories.
- Separate alcohol into liquor, beer, wine and other. They have different cost structures, different tax treatment in many jurisdictions, and different benchmarks. Merging them makes beverage cost analysis useless.
- Keep non-revenue items (gift card sales, deposits, service charges) out of your food and beverage categories entirely. Gift card sales are not revenue at the point of sale; they are a liability until redeemed.
- Decide where modifier revenue lands. Add-ons that carry a price should report into a category you can see, not vanish into the parent item.
- Map every old category to a new one and keep the mapping table. You will need it the first time you compare this year to last year.

Modifier groups: where rebuilds actually fail

More switch problems come from modifiers than from anything else. Work through each trap below deliberately.

Trap	What goes wrong	How to build it correctly
Required versus optional	A required choice built as optional means tickets reach the kitchen without a temperature, a side or a protein choice, and the line has to chase the server.	Set required groups with a minimum of 1. Test by trying to send the item without choosing - the system should stop you.
Minimum and maximum selections	A "choose two sides" group with no maximum lets a guest be charged for two and receive four. A maximum set to 1 on a group that should allow three blocks a legitimate order and the server rings an open item instead.	Write the min and max for every group on paper before you build. Test the boundary: try to select one too few and one too many.
Price-carrying modifiers	Add-ons priced at zero because the builder assumed the price came from somewhere else. Extra protein goes out free for months before anyone notices.	List every modifier that should carry a price and its amount. After the build, ring a ticket with each one and check the cheque total.
Upcharge logic by context	The same add-on should cost different amounts on different items - extra cheese on a burger versus on a large pizza. Systems handle this very differently.	Identify every context-dependent price and decide: separate modifier per context, or a size-aware price. Do not rely on staff to remember.
Nested and dependent modifiers	Choosing one option should reveal another (choose a sandwich, then a bread, then a toast level). Flat modifier lists produce long screens and slow tickets.	Build the chain and time a server ringing it. If it takes more than a few taps, restructure.
Size and variant handling	Sizes built as separate items break product mix reporting; sizes built as modifiers can break recipe costing. Both are defensible - inconsistency is not.	Pick one approach and apply it across the whole menu. Document the choice.

Trap	What goes wrong	How to build it correctly
Combos and meal deals	Combos that discount the parent item make component-level sales invisible, so you cannot see which items are actually selling.	Build combos so components still report into their own categories, with the discount recorded as a discount rather than a price change.
Default selections	Defaults that are pre-selected get sent unnoticed - the guest who did not want the sauce gets it, and the remake is on you.	Use defaults only where the answer is genuinely default (a burger comes with fries). Never default a paid add-on.
No-modifiers and allergen flags	"No nuts" typed into an open note is not a control. Free-text notes do not reliably print in a way the line will see.	Build explicit allergen and no-ingredient modifiers that print prominently on the kitchen ticket or highlight on the KDS.
Open items and miscellaneous buttons	Unrestricted open-price buttons are the most common internal control weakness in a POS and destroy product mix reporting.	Restrict to manager permission, require a description, and review usage weekly.
Modifier print routing	Modifiers that do not follow the item to the correct printer or screen.	Test by ringing one item with every modifier group and reading the physical ticket at the station.

Menu build checklist

- ☐ Prune before you build. Pull the product mix report from the old system, sort by units sold, and question everything in the bottom quartile. Rebuilding items you are about to delete is wasted work.
- ☐ Build categories first, then modifier groups, then items. Building items first means editing every one of them later.
- ☐ Name items the way staff speak, not the way the menu prints - servers search by the word they use.
- ☐ Set the kitchen-ticket name separately from the guest-facing name where the system allows it. The line wants "CHKN SNDWCH", the guest wants the full description.
- ☐ Assign every item to a print or KDS route as you create it, not in a second pass.
- ☐ Enter costs against items if the system supports it, so theoretical food cost is available from day one.
- ☐ Ring every single item at least once on a live terminal before go-live. Not a sample - every item. Two people, one afternoon.
- ☐ Print the full item list with prices and have the chef and the owner sign it off against the physical menu.
- ☐ Freeze the build one week before training. Nothing changes after the freeze except genuine errors, and every change is logged.

Tax and tender configuration

- ☐ Document every tax rule that applies to you in plain language before you touch the system: which categories are taxable, at what rate, and whether dine-in, take-out and delivery differ.
- ☐ Confirm the treatment of alcohol, which is frequently taxed differently.
- ☐ Configure tax-exempt handling for any accounts entitled to it, and restrict who can apply an exemption.
- ☐ Ring one test ticket for every distinct tax case and check the cent-level result against a hand calculation. Keep the receipts.
- ☐ Configure every tender type you actually accept: cash, each card type, gift card, house account, third-party delivery, and any local payment method. Split them so reconciliation is possible.
- ☐ Set up third-party delivery as its own tender or its own revenue centre so marketplace sales never contaminate your dine-in averages.
- ☐ Configure tip handling: suggested percentages, whether they calculate on pre-tax or post-tax totals, automatic gratuity thresholds, and how tips flow to payroll.
- ☐ Configure service charges separately from tips, and confirm the tax treatment of service charges in your jurisdiction - it often differs from tips.
- ☐ Configure rounding rules if your currency requires cash rounding.
- ☐ Set discount and comp reason codes, and make a reason mandatory on every one.
- ☐ Set the void policy: what can be voided without a manager, and what always requires one.

Floor plan, routing and hardware

Floor plan and table mapping

- ☐ Draw the floor plan to match the physical room, including the bar, the patio and any private area. An abstract grid slows servers down.
- ☐ Keep the existing table numbers if staff already know them. Renumbering a room during a system switch adds an unnecessary source of errors.
- ☐ Set seat positions on tables if you use seat-level ordering, and confirm the kitchen ticket prints seat numbers.
- ☐ Create revenue centres that match how you want to read sales: dining room, bar, patio, take-out, delivery, catering.
- ☐ Set table capacities correctly so any waitlist or reservation integration seats parties sensibly.
- ☐ Test a table transfer, a cheque split by item, a cheque split evenly by four, and a merge of two tables. These are the four operations servers will need on night one and they are the ones most often untested.

Printer and KDS routing matrix

Fill this in before the hardware is installed, then test every row with a live ticket walked to the station. The backup column is the one everybody skips and the one that saves a Friday.

Station	Items routed	Printer or screen	Backup route	Tested
Hot line / grill	Mains, grilled items, hot appetisers			
Saute / range	Pasta, sauced dishes, risotto			
Fry	Fried items and fried sides			
Cold / garde manger	Salads, cold appetisers, raw bar			
Pizza / oven	Pizza, flatbread, baked items			
Pastry / dessert	Desserts, plated sweets			
Expo	Full ticket, all courses, all modifiers			
Bar	All beverages, alcohol and non-alcohol			
Service bar	Beverages for the dining room only			
Take-out / packing	All take-out and pickup orders			
Delivery / third party	Marketplace and direct delivery orders			
Front counter / receipt	Guest receipts and cheque presentation			
Manager / office	Reprints, void and comp slips, reports			

Routing rules that prevent lost tickets

- Every station must have a defined backup. If the grill printer fails, the ticket should reprint at expo, not vanish.
- Configure the system to alert when a print job fails. Silent failures are the worst possible outcome, because service continues while food is never made.
- Decide what happens to a modifier when the item routes to two stations - it should follow the item, not the ticket.
- Print course markers and fire timing on the ticket if you run coursed service.
- Keep one spare printer, one spare cable and one spare paper roll box on site. Thermal printers fail on the busiest night, not the quietest.

Hardware and network

- ☐ Hardwired network drop at every terminal, KDS and kitchen printer. Wireless is a backup, not a plan.
- ☐ Business-grade router, separate staff and guest networks, and a cellular failover.
- ☐ Every terminal, switch, router, KDS and kitchen printer on an uninterruptible power supply.
- ☐ Confirm power outlet locations before install day. Extension cords across a kitchen floor are a safety and inspection problem.
- ☐ Label every cable at both ends. You will be grateful at 7 pm on a Saturday.
- ☐ Stage and configure hardware before install day so go-live is a physical task, not a configuration task.
- ☐ Keep the old hardware powered and connected until the new system has run a clean week - see the cancellation section.

The payments cutover

Payments are the part of a switch that touches real money and real regulation. Treat it as its own project with its own owner.

Before cutover

- ☐ Complete merchant underwriting early. A new merchant account can take one to three weeks and may require financial statements, identification for beneficial owners, and a personal guarantee.
- ☐ Confirm your funding schedule in writing: what time the batch closes, how many business days until funds land, and what happens on weekends and holidays.
- ☐ Confirm the descriptor that will appear on guest statements. A wrong or unrecognisable descriptor generates chargebacks.
- ☐ Get the full fee schedule, not a rate: interchange treatment, assessments, monthly minimums, statement fees, PCI fees, chargeback fees, batch fees, gateway fees.
- ☐ Confirm the terminals are ordered, and confirm whether they are countertop, handheld or pay-at-table, and how many. Order at least one spare.
- ☐ Confirm how card data is handled. In a modern setup, card data is captured by the payment terminal or the provider's hosted page and never touches your POS or your network - that is what keeps your PCI obligations light. Ask the vendor to state this in writing.
- ☐ Confirm tip flow end to end: how a tip is added, whether tips are adjusted after authorisation or captured at the terminal, and how tips reach payroll.
- ☐ Test a sale, a partial refund, a full refund, a void before batch, a tip adjustment and a declined card on the new terminals before go-live.
- ☐ Confirm what happens to cards presented while the internet is down, including any offline floor limit and who carries the risk of a later decline.
- ☐ Confirm chargeback handling: who receives the notice, what evidence is required, and how long you have to respond.

Cutover night

- ☐ Settle and batch out the old processor before you switch. Do not leave an unsettled batch on the old system overnight.
- ☐ Print and keep the final settlement report from the old system.
- ☐ Physically disconnect old terminals so no one can accidentally take a payment on them, but do not return them yet.
- ☐ Run one live card transaction on each new terminal - a real card, a real small amount - and refund it. Do not go live on an untested terminal.
- ☐ Confirm the first batch closes correctly at the scheduled time.

Reconciling the first week

For the first seven days, reconcile every day, by hand, before you trust any report. This is how you find configuration errors while they are still cheap.

Compare this	Against this	What a mismatch usually means
POS net sales by tender type	POS daily closeout total	A tender type is mis-mapped, or a revenue centre is excluded from the report.
POS card totals by card brand	Processor batch report	A terminal batched late, a tip adjustment landed after the batch, or a transaction was run on the wrong device.
Processor batch report	Bank deposit on the funding date	Fees are being netted from the deposit (normal - confirm the amount), a batch did not close, or funding is on a different schedule than you were told.
POS tips total	Processor tips total and payroll tip import	Tip adjustments made after batch, or a tip-pool rule configured differently in the new system.
POS cash sales less paid-outs	Actual cash counted and deposited	Drawer procedure, paid-out recording, or a genuine cash control problem.
POS tax collected by rate	Hand calculation from category sales	A tax rule is wrong. Find this on day one, not at filing.
Gift card sales and redemptions	Gift card liability register	Cards issued or redeemed outside the system, or migration balances not loaded.
Third-party delivery sales in POS	Marketplace dashboard totals	Integration duplication or omission - the single most common source of overstated sales after a switch.

Training the team

Train managers first and separately. A manager who is still learning the system cannot support a server during service, and the whole floor watches how confidently management handles it.

Role	Hours	Trained by	Must be able to do without help
General manager and owner	6-8	Vendor implementation specialist	Full back-office: menu edits, price changes, employee and permission management, every report, void and comp review, end-of-day and reconciliation.
Assistant managers / supervisors	4-6	GM plus vendor	Void, comps, discounts, cheque reopens, tip adjustments, drawer counts, end-of-day, and how to fix the five things servers most often get stuck on.
Servers	2-3	GM or a designated lead server	Open a table, ring items with every modifier type, split a cheque by item and evenly, transfer a table, apply a discount, take payment, add a tip, close, reprint a receipt, process a refund with manager approval.
Bartenders	2-3	Bar lead	All server tasks plus opening and transferring tabs, pre-authorising a card, closing a tab to a table, and bar-specific modifiers.
Hosts	1-2	GM	Seat a table, manage the waitlist and reservations, take a take-out order, process a payment.
Kitchen - line and expo	1-2	Chef plus vendor	Read a ticket or KDS screen, bump and recall an order, mark an item eighty-sixed, and know what to do when a printer or screen fails.
Take-out and delivery staff	1-2	GM	Take and modify a phone order, handle a third-party order, quote realistic times, process refunds.
Everyone	0.5	GM	Clock in and out, view their own schedule and hours, and report a problem through the agreed channel.

Training that actually sticks

- Train on the real system with the real menu, in the real room, on the real terminals. Training on a laptop in the office teaches nothing about where the buttons are when you are standing at station three.
- Use scenarios, not screen tours. "Table 12, four guests, one allergy, two separate cheques, one paying half cash" teaches more than an hour of demonstration.
- Have every server ring a minimum number of practice tickets and sign off. Twenty is a reasonable floor.
- Print a one-page cheat sheet for the ten most common tasks and tape it beside every terminal for the first month.
- Identify two or three super-users per shift - usually the fastest learners, not the most senior - and tell the team to ask them first.
- Run a dress rehearsal: a full simulated service on the new system, with real ticket times measured and a debrief afterwards.
- Agree the escalation path in advance: server asks a super-user, super-user asks a manager, manager calls the vendor. Write the vendor's number on the cheat sheet.

Parallel run or big bang?

Both are legitimate. Choose deliberately, in week 2, and tell the team which one you have chosen.

	Big bang (hard cutover)	Parallel run
What it means	Old system off, new system on, at a single moment - usually after close on your slowest night.	Both systems live for a defined period. Typically the new system runs one revenue centre, one day-part or one station while the old system runs the rest.
Best for	Single-location restaurants with a straightforward menu, and any operation where double-entry would slow service unacceptably.	High-volume or complex operations, multi-revenue-centre sites, and anyone whose menu has heavy modifier logic that must be proven under real conditions.
Advantages	Clean data boundary; no double entry; no staff confusion about which system to use; simpler reconciliation; the team commits fully.	Real-world validation before full commitment; a genuine fallback; staff learn under lower pressure; problems surface on a small blast radius.
Risks	Problems appear at full volume with no fallback. Requires a genuinely tested rollback plan and vendor support on site or on call.	Split reporting across two systems for the period; double the reconciliation work; risk that staff quietly retreat to the old system and never learn the new one.
Make it work by	Cutting over on your slowest night, having vendor support present, keeping the old system powered and available for one week, and reducing the menu for the first two services.	Setting a hard end date for the parallel period - two weeks maximum - and publishing it. Parallel runs without an end date become permanent.
Do not choose this if	You have no rollback plan, no vendor support on the night, or a menu that has not been fully ring-tested.	Your team is small enough that double entry will visibly slow service, or your reconciliation capacity is one part-time bookkeeper.

COMMON MISTAKE

Cutting over on a Monday morning "because it is quiet" and then discovering the first real test is Friday night, four days later, with the vendor's implementation team already reassigned. Cut over so that your first busy service falls while support is still contractually engaged, and put that expectation in the contract before you sign.

The cutover runbook, hour by hour

This assumes a Sunday-night cutover for a Monday opening. Shift the clock to your own schedule but keep the sequence and keep the named owners. Print this page and tick it off on the night.

Time	Task	Owner	Done
Friday	Final data export from the old system; verify files open; store two copies.	GM	
Friday	Confirm vendor support attendance in writing; confirm the mobile number of the person who will answer.	GM	
Saturday	Print cheat sheets, cutover runbook and full item list with prices. Put them in the office.	GM	
Sun 21:00	Last order taken on the old system.	Manager on duty	
Sun 22:00	Close all open cheques on the old system. No cheque carries over.	Manager on duty	
Sun 22:15	Settle and batch out the old payment processor. Print the settlement report.	Manager on duty	
Sun 22:30	Run every closing report on the old system: sales, product mix, labour, voids and comps, cash, gift card balances. Print and file.	GM	
Sun 22:45	Final incremental data export from the old system covering the last trading days.	GM	
Sun 23:00	Count and reconcile cash. Zero the drawers.	Manager on duty	
Sun 23:00	Photograph the old terminal screens: key layouts, modifier screens, any configuration not exported.	Assistant manager	
Sun 23:15	Old system powered down but left connected and available. Do not uninstall. Do not return hardware.	Vendor / GM	
Sun 23:15	New terminals mounted, cabled and powered. All devices on UPS.	New vendor	
Sun 23:45	Network verified on every device; failover tested by unplugging the primary connection.	New vendor	
Sun 00:00	Menu final verification: two people ring one item from every category and every modifier group.	GM + chef	
Sun 00:15	Print a live test ticket to every station and physically walk each one to its station.	Chef	
Sun 00:30	Live card test on every payment terminal: one sale, one refund, one tip adjustment.	GM	
Sun 00:45	Employee records, PINs and permissions verified. Every manager logs in successfully.	GM	
Sun 01:00	Gift card and loyalty balances loaded or manual register printed and placed at the host stand and the bar.	GM	
Sun 01:15	Floats counted into new drawers, two signatures.	GM	
Sun 01:30	Cheat sheets taped beside every terminal. Vendor support number written on each one.	Assistant manager	
Mon 09:00	Pre-shift briefing: what changed, where to find help, the reduced menu for today, and permission to be slower than normal.	GM	
Mon 10:30	Open with a reduced menu and, if possible, reduced hours. Vendor support present or on call.	All	
Mon 11:00	First live guest transaction watched end to end by a manager, from ring to kitchen to payment to receipt.	GM	

Time	Task	Owner	Done
Mon 15:00	Mid-day checkpoint: ticket times, staff issues, any item ringing incorrectly. Fix and log.	GM + chef	
Mon close	Full end-of-day; reconcile POS to processor to cash; note every discrepancy.	GM	
Tue 09:00	Confirm the first bank deposit matches the first batch. Investigate any gap the same day.	GM + bookkeeper	
Tue 09:30	Debrief with the team. Collect the top five friction points and send them to the vendor as one list.	GM	

Day 1 to day 7 stabilisation

- ☐ **Day 1:** reduced menu, extra manager on the floor, vendor reachable. Watch the pass and the payment terminal, not the office.
- ☐ **Day 1:** reconcile POS to processor to cash before anyone goes home.
- ☐ **Day 2:** confirm the first deposit landed and matches. Confirm tips flowed to payroll correctly.
- ☐ **Day 2:** collect friction points from staff in one written list. Send the list to the vendor once, not in seven separate messages.
- ☐ **Day 3:** restore the full menu if ticket times are stable.
- ☐ **Day 3:** run every report you rely on and check each one against a hand calculation or the old system's history. Reports are where quiet configuration errors hide.
- ☐ **Day 4:** review voids, comps and discounts by employee. A spike is almost always confusion, not dishonesty - retrain before you accuse.
- ☐ **Day 4:** check product mix. Items showing zero sales are usually mis-built or unreachable on the screen, not unpopular.
- ☐ **Day 5:** button layout pass. Move the ten most-used items to where staff actually reach for them, based on a week of watching.
- ☐ **Day 5:** verify tax collected by rate against a hand calculation for the week.
- ☐ **Day 6:** reconcile gift cards and loyalty for the week. Confirm every old card presented was honoured and recorded.
- ☐ **Day 7:** full weekly reconciliation: sales, labour, payments, cash, tax. Compare the week to the same week on the old system.
- ☐ **Day 7:** written go/no-go decision on cancelling the old contract. Only cancel once the data is verified and the week reconciles.

Signals that you are not stable yet

- Cash variance still outside your normal tolerance after five days.
- Any report that you cannot reconcile to a hand calculation.
- Staff routinely using an open-item or miscellaneous button to work around a menu item that does not ring correctly.
- Kitchen tickets arriving without modifiers, or arriving at the wrong station.
- Ticket times still materially above your pre-switch baseline.
- Any unexplained gap between the processor batch and the bank deposit.

If any of these are true, do not cancel the old contract this week. The cost of one more month of the old subscription is trivial compared with losing your only route back.

Cancel the old contract properly

More money is lost in the last two weeks of a switch than in the first four. Work this section in order, and do all of it in writing.

Sequence

1. **Read the contract before you tell anyone you are leaving.** Find the term end date, the notice period, the notice method (many contracts require written notice by post or to a specific address), and the auto-renewal clause.
2. **Diarise the notice date.** Evergreen contracts commonly renew automatically unless you give notice inside a specific window - often 30 to 90 days before the term end. Miss the window by one day and you owe another full term.
3. **Calculate the cost of leaving early** if you are mid-term: early termination fee, unamortised hardware, remaining months. Get it in writing and check the arithmetic against the contract yourself.
4. **Take the final data exports** - all of them, verified - before you send notice. Access can change the moment a cancellation is logged.
5. **Send notice exactly as the contract requires**, on the required date, by the required method, and keep proof of delivery.
6. **Ask for written acknowledgement** of the cancellation, the effective date, and the final amount owing.
7. **Cancel any linked services separately** - payment processing, online ordering, loyalty, gift cards, reservations, reporting add-ons. These are often separate agreements with separate terms and they do not cancel automatically.
8. **Cancel the payment processing agreement in writing** and confirm the final funding date. Keep the merchant account open until the last batch has funded and any chargeback window you care about has passed.
9. **Return hardware only when instructed**, to the address given, with tracking, and photograph everything before it ships. Missing-equipment charges are common and are hard to dispute without photographs and a tracking number.
10. **Wipe or confirm the wipe of any device holding guest data** before it leaves your premises.
11. **Check the final invoice line by line** against the contract: prorated subscription, any early termination fee, hardware charges, and confirm that recurring billing has actually stopped.
12. **Cancel the direct debit or card on file only after the final invoice is settled** - cancelling first can trigger a default under the contract.
13. **Watch your bank statement for three months** for any recurring charge that did not stop.

DO THIS BEFORE YOU SIGN ANYTHING NEW

Ask the new vendor for the same three things you wish you had asked the old one: (1) a written self-serve data export covering menu, sales history, guests and loyalty balances, available at any time without their assistance; (2) the exact notice period and a commitment that the contract does not auto-renew for a full term; and (3) the total cost in month one, month thirteen and month twenty-five, itemised, including payment processing on your real card mix. If a vendor will not put those three in writing, you already know what leaving them will feel like.

Questions to ask the new vendor

Ask these during evaluation, not after signature. Record the answers and attach them to the contract file.

Area	Question	What a good answer sounds like
Cost	What is my all-in monthly cost in month 1, month 13 and month 25, itemised?	An itemised quote with every module, per-terminal fee and hardware cost listed, plus the contracted escalation percentage.
Payments	What is my effective processing rate on this card mix, including every fixed fee?	A blended effective rate calculated on a mix you supplied, plus a written fee schedule with no unexplained line items.
Payments	Is card data ever stored on my POS or my network?	No - captured by the terminal or a hosted page, tokenised, never on your hardware. Ask them to state it in writing.
Payments	Am I locked into your processing, and what does it cost to change?	A clear answer either way. A vendor who will not answer this directly is telling you the answer.
Contract	What is the term, the notice period, and does it auto-renew?	A stated term, a reasonable notice window, and no evergreen renewal for a full additional term.
Contract	What is the early termination fee and how is it calculated?	A formula you can compute yourself, not a discretionary figure.
Data	Can I export menu, sales history, guests and loyalty balances myself, at any time?	Self-serve CSV or API export of all four, without raising a support ticket.
Data	How long do I retain access to historical data after cancellation?	A stated period in the contract, not an informal assurance.
Menu	Build my most awkward menu item live, right now.	They do it in the demo, in under ten minutes, including the required choice, the priced add-on and the size variant.
Resilience	Show me the system with the network cable pulled.	Orders still ring, tickets still print, cards still process within a stated offline limit, and everything syncs when connectivity returns.
Hardware	Who owns the hardware, what does replacement cost, and what is the turnaround on a failure?	A named replacement path with a stated turnaround, and the option to hold a spare.
Implementation	Who builds my menu, how many hours of training are included, and who is on site at go-live?	Named deliverables, named hours, and a named person for the cutover night.
Support	What are your support hours and how do I reach a human on a Saturday night?	A phone number you can call during evaluation - and a human answers when you test it.
Reporting	Show me the six reports I will run every Monday.	Sales by category, labour against sales by day-part, voids and comps by employee, discounts by reason, product mix with margin, and payment reconciliation.
Integrations	Which accounting, payroll, delivery, reservation and inventory systems do you integrate with, and at what cost?	A current list with named partners, and the fee for each, in writing.
Roadmap	What has shipped in the last twelve months?	Specific, dated releases. Roadmap promises are not deliverables.

One last thing

The best switches feel anticlimactic. If your cutover night is dramatic, something was skipped in week 5 - almost always the data extraction, the modifier testing, or the decision about who owns the project. Work those three hard and the rest of this playbook becomes a formality.