Group or chain

Many properties, one way of working

Playbook | For: VPs of operations, regional managers, group IT and owners of hotel groups

Who this is for

You run a group of properties. Picture a 12-hotel group with 3,000 rooms: a few large city hotels, several mid-size properties and a couple of remote lodges. You want every property working the same way, and you want one view of how they are doing.

The limit, stated plainly

ExpTix is one property per install today. A group runs one install per property. Multi-property in one install is on the roadmap. It is not built.

So, today:

- each property has its own address, database, users and settings,
- a person who works at two properties has an account on each,
- there is no group dashboard inside ExpTix,
- settings are not shared or copied between installs automatically.

Everything below is how to run a portfolio well inside that limit. When multi-property support arrives, a group that has already standardized its settings will have far less to clean up.

The model: a gold template property

Pick one install as your **gold template**. This is where the group’s standard way of working lives. It can be a real property that runs well, or a non-guest install used only for standards.

The gold template holds:

Setting	What to standardize
Departments	Same codes and names everywhere (for example HK, ENG, FO, KIT). Properties switch off what they do not have, but never rename a code.
SLA policies	Group standard timers per priority, plus the guest amenity and complaint promises.

Setting	What to standardize
Request categories	Same names, ticket types, default priorities and auto-route keywords. Properties may add local ones.
Checklist templates	Brand room inspection, opening and closing checks, fire and life safety rounds, HACCP.
PM schedules	Standard tasks and frequencies per asset type.
Notification rules	The group pattern: who hears about what, at which minimum priority.
Room types	Same codes and cleaning credits for the same kind of room.

Copying the template to a new install

There is no button that clones settings from one install to another. Use one of two ways:

1. **Manual copy.** Recreate the settings in the new install's admin screens, working from a printed or exported list of the gold template. Fine for a handful of properties.
2. **A script against the REST API.** The API's Records endpoints (see `/docs` on any install) list and create departments, SLA policies, categories, checklist templates, PM schedules and notification rules. A short script reads them from the gold install and creates them on the new one. Record ids differ between installs, so the script must map references (for example a category's department and SLA policy) by code or name, not by id. Use a dedicated user and a token with the right scopes, and delete the token when the job is done.

Whichever way you choose, write down what was copied and when. Keep the script in your own version control.

Naming and numbering conventions

Set these once at group level and never change them per property.

- **Property code.** Each install sets `property_code`, which prefixes every ticket number. Use a short, unique code per property, for example `DTN` for a downtown hotel and `LKS` for a lakeside resort. Tickets then read `DTN-000123` and `LKS-000123`, so nobody confuses a job from one hotel with another in email, reports or BI.

- **Property name, time zone, currency.** Set `property_name`, `timezone` and `currency` correctly on each install.
- **Department codes.** Identical across properties, as above.
- **Room types.** Same code for the same product.
- **Locations.** One pattern for building, floor and area names.
- **Usernames.** One pattern for the whole group, so a person's name is easy to find in any install's **Audit log**.
- **Web addresses.** One pattern, for example a sub-domain per property code.

Central reporting

There is no group report inside ExpTix. Build one outside it.

1. **CSV export.** Each property's **Reports** page exports tickets to CSV (up to 500 per download). Good for a monthly look, not for a feed.
2. **REST API into a BI tool.** For each property, create a dedicated read-only user (the Auditor role sees every department and changes nothing) and a token with only the `read` scope and an expiry. Your BI tool pulls tickets from each install, 500 per page with `limit` and `offset`, and tags each row with its property code. The code is already in every ticket number.
3. **Outbound webhooks.** Each install can send events to an HTTPS endpoint you run. Every payload carries the property code. Verify the signature.

Build the same few measures for every property, from the same fields: response and resolve times, SLA misses by priority, guest ratings, complaints and recovery value, repeat-problem rooms, PM on time. Compare properties of a similar size and type, and compare each property with its own baseline. Do not publish savings figures before you have measured them.

The regional admin model

Level	Who	What they own
Group	VP of operations, group IT	The gold template, conventions, change control, BI, security standards
Region	Regional administrator (one person can cover several properties)	Installs, upgrades, backups, integrations and users for their properties
Property	GM and a local champion	Daily use, local categories, training, feedback

The regional administrator holds an Administrator account on each property in their region, with MFA forced. Property GMs manage their own staff in **Users**. Group leaders who want to look but not touch get the Auditor role on each install.

Rollout waves

Wave	Properties	Purpose
0	The gold template, on a test install	Build and test the standard settings
1	One mid-size property with a strong GM	Prove the template in real use. Follow the rollout plan
2	Two or three similar properties	Prove the copy process and the training
3 onward	The rest, in groups of similar properties	Steady rollout

Large properties follow the large hotel playbook, with a department-by-department pilot and a load test. Small or remote ones follow the small lodge playbook and may switch off most departments.

Feed what each wave learns back into the gold template before the next wave starts.

Governance and change control

Settings drift unless someone owns them.

1. **One owner for the template.** Changes to group-standard settings go through them.
2. **A simple change request.** What changes, why, which properties, who approves.
3. **Test, then copy.** Change the gold template first, check it on the test install, then copy it out with the same script or checklist.
4. **Local changes are allowed, within limits.** Properties may add categories or checklists. They may not change department codes, standard SLAs or group notification rules without approval.
5. **Same version everywhere.** Upgrade the test install first, then all properties in a short window. Read `CHANGELOG.md` each time.
6. **Quarterly review.** Compare each install’s settings with the template and fix drift.
7. **Security standards.** HTTPS, MFA for managers, daily off-server backups, `app.key` stored separately, the audit webhook to your log platform, a regular **Verify integrity** check.

Pitfalls

- **Starting all properties at once.** You learn nothing before the mistakes are everywhere.
- **Letting each property invent its own categories.** Reports cannot be compared.
- **Shared admin passwords across installs.** Each administrator has their own account on each install.
- **Expecting single sign-on.** It is on the roadmap (Entra ID, Google), not built. Plan on separate passwords with MFA.
- **Expecting the app in other languages.** The screens are English only today. Translate printed guides now.
- **No support plan.** ExpTix is free software with no vendor support contract. Group IT or a chosen IT provider must own it.