



---

## **SAP Role Naming Convention: Do we need one?**

**Author: Andy Hartley**

**Tango Technologies Ltd (TangoTec)**

---

### **A moment at your desk**

You're at your desk, staring at a blank naming field and wondering whether a role naming convention really matters. You've asked yourself the same question I've heard a thousand times: if role names must be unique, why fuss over the convention? The short answer is that a good convention saves time, prevents mistakes, and looks very attractive to auditors.

---

### **A few real-world naming culprits**

Over 30 years I've seen every flavour of naming convention. Here are a few that make you wince:

Numeric-only names: Z0000001, Z0000002, Z0000003. The idea was to force security teams to inspect role content rather than rely on names. Clever on paper; disastrous in practice. It solves a non-problem and makes life harder for everyone.

Org-level in the name: putting a single code like 1000 into the name for plant or company code. Ambiguous, brittle, and a nightmare when an employee spans multiple org values. I've pushed clients away from this more than once.

No convention at all: "We just make names up as we go." This is the one that gets auditors reaching for their tear-off notepads

---



## **What a usable convention needs to do**

A practical convention should:

- Make intent obvious at a glance.
- Prevent accidental assignment of dangerous roles.
- Support transport rules across Dev → QAS → Prod.
- Play nicely with SAP authorization checks like S\_USER\_SAS and S\_USER\_GRP.
- Be policy-driven so it's repeatable and enforced.

---

## **My recommended structure (simple, policy-driven)**

I like a compact, readable pattern that encodes assignability, system scope, role type, and business area:

- First character — assignability: Y = master role (not assignable), Z = assignable role.
- Second character — system scope: D = Development only, Q = Quality, P = Production assignable. This controls how far a role can travel in the transport path.
- Third character — role type: S = single/standalone, D = derived, C = composite.
- Fourth character — separator: underscore (\_) for readability.
- Remainder — business area and department: e.g., LON\_FIN for London Finance.

Examples:

ZPC\_LON\_FIN... — a composite role assignable in Production for London Finance.

ZQD\_WOR\_HCM... — a derived role assignable in Quality for HR testing for Worcester Users.

YDM\_XXX\_SAL... — a non assignable master role only to exist in Development, used to derive regional sales Roles.



## Why this helps with control and automation

By aligning names to policy, you can tie system controls to naming and user group membership. For example, when end-user groups are created as END\_LON\_FIN, the S\_USER\_SAS authorization object can enforce that a user in END\_LON\_FIN only receives roles that belong to that area. That prevents someone from assigning an inappropriate global finance role even if it's been approved. The naming convention becomes part of the control framework rather than just a cosmetic label.

---

## Practical tips for rollout

- Enforce via role creation workflows or simple validation scripts to stop bad names at source.
- Use the naming to drive transport and assignment rules so Dev-only roles never end up in Prod.
- Keep the business-area suffix clear and consistent; use agreed abbreviations and a short registry.
- Make master roles (Y\*) and assignable roles (Z\*) visually distinct in role catalogues and dashboards.

---

## Final note

This convention is intentionally pragmatic: it balances readability, enforceability, and integration with SAP authorization objects. It reduces accidental mis-assignments, speeds audits, and forces conversations about what a role actually does rather than what it's called.