Active Directory Structure Guidelines – Part 1

Posted by Alan Burchill on 23 July 2010, 8:00 pm

I have been doing Active Directory and Group Policy work for a while now and I have developed my own set of rules that I try to use where ever possible. So below I have written down all my rules in no particular order for you to go over and use for yourself. You may only chose to use only some of these rules or you might want to use them all depending on your circumstance. This is a two part series where I will first talk about designing you Active Directory Organisation Unit structure and then in part 2 (Best Practice: Group Policy Design Guidelines – Part 2) I will discuss some more ideas for applying Group Policy to the OU structure.

I want to be clear that these are only guidelines and not rules that need to be strictly adhered to. In almost all case there are exceptions to these guidelines and you might even find your self implementing them in a hybrid approach. I intend for this web page to be updated on a regular basis as none of these rules are set in stone and thing obviously change all the time.

Active Directory Organisation Unit Design Guidelines
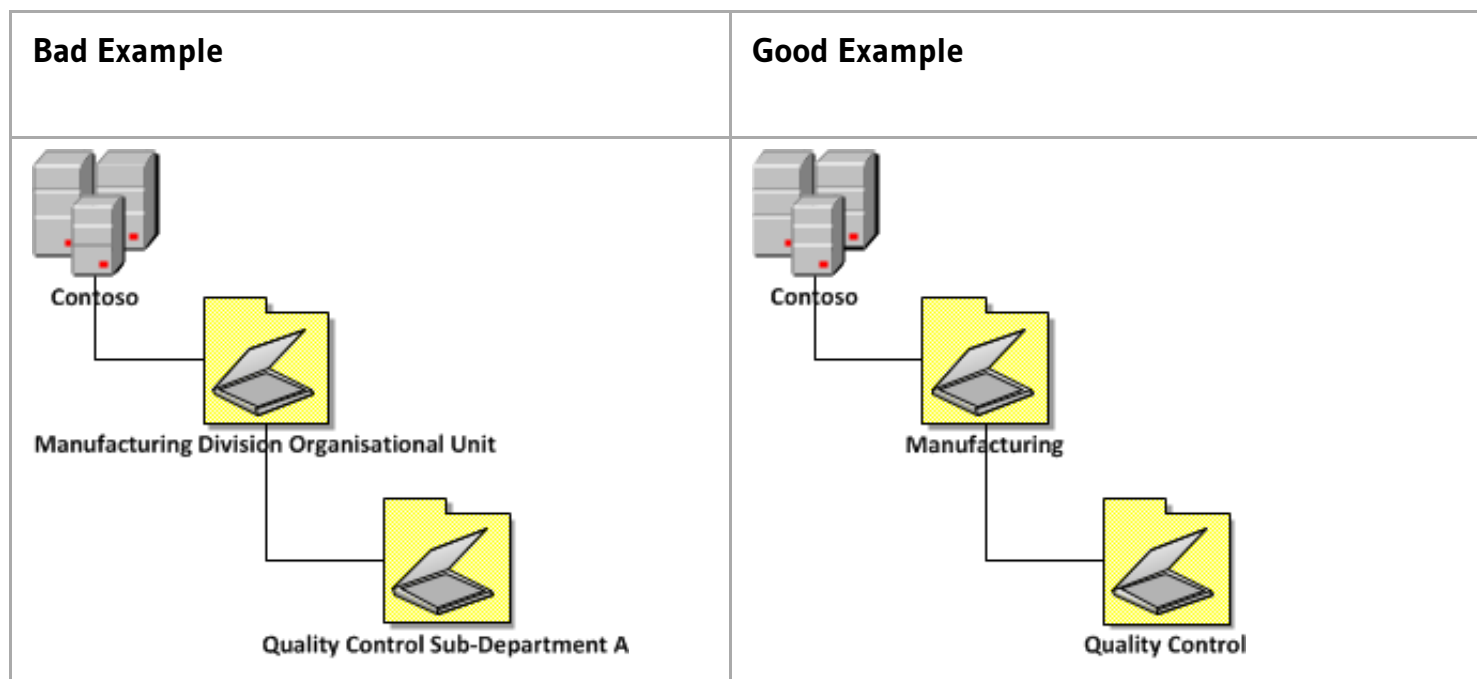
Before you begin

Before you begin the process of designing your Group Policy (and AD) structure you should first try to fully understand the requirements of the environment. Below are some points that I recommend that you find out before you begin:

- How is the company structured
- Where are the physical sites
- Who support the organisation
  - What are the support boundaries (e.g. Location and/or Workstations and/or Servers)
- What are the computer types
  - Highly Secured
  - Standard SOE

- Process Control/Automation
  - Server Roles (e.g. Exchange, SQL or File Server)
- Network Topology
  - Bandwidth / Latency
- Who will be responsible for Group Policy changes
- What are the security requirements (password policy, auditing etc.)
- What is the change management process
- What are the auditing requirements for Group Policy
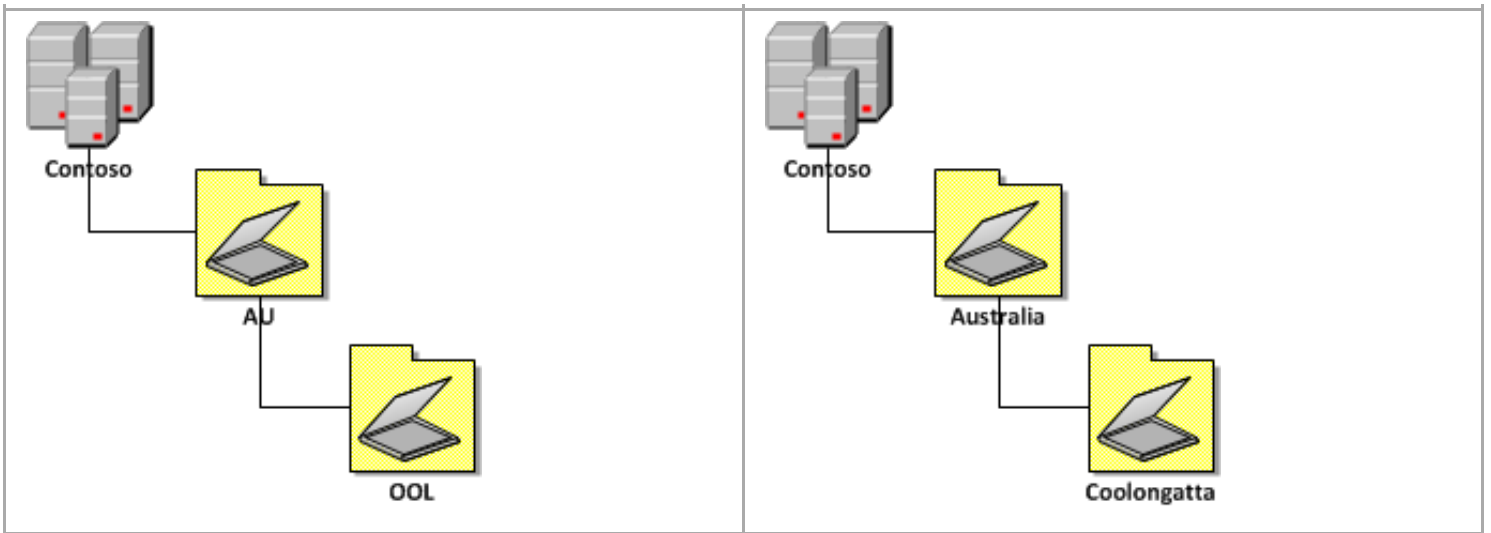
Keep it short

When naming your Organisational Unit make sure the name you are using are short and to the point. There is technically nothing wrong with having long OU names but it is a pain to document and just leave you open to more chance of references then name wrong as their are more characters to type.

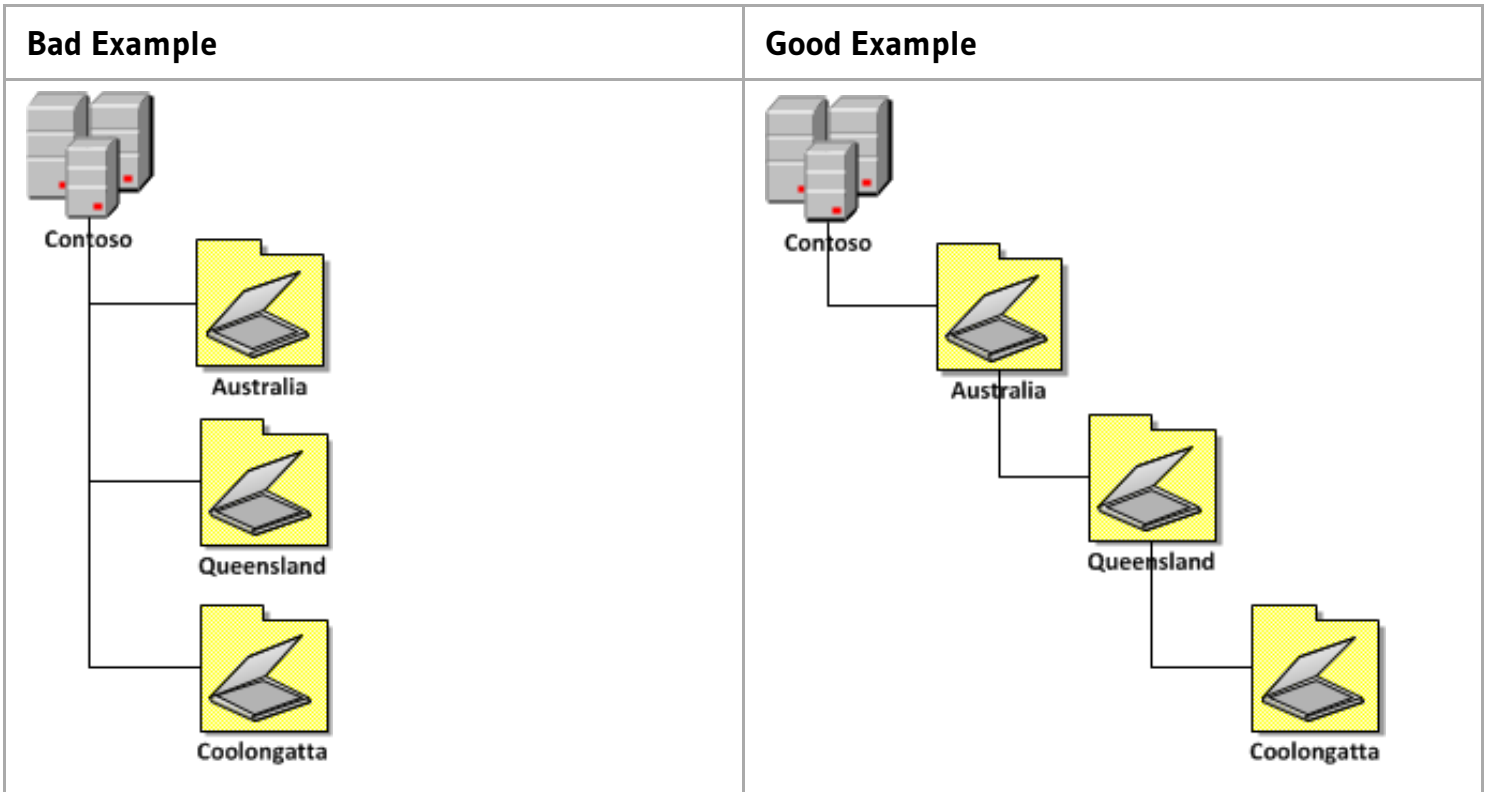| Bad Example	Good Example
 Contoso Manufacturing Division Organisational Unit Quality Control Sub-Department A	 Contoso Manufacturing Quality Control

Be intuitive

Naming OU to something that is intuitive is good for new starters in the organisation. If you name a OU "OOG" a new starter in your organisation might not realise that this is the three letter international designation for Coolangatta AirPort which is the same suburb where your office is located. I know this is in conflict with rule 1 however it is also a balancing act your will have to carefully tread.

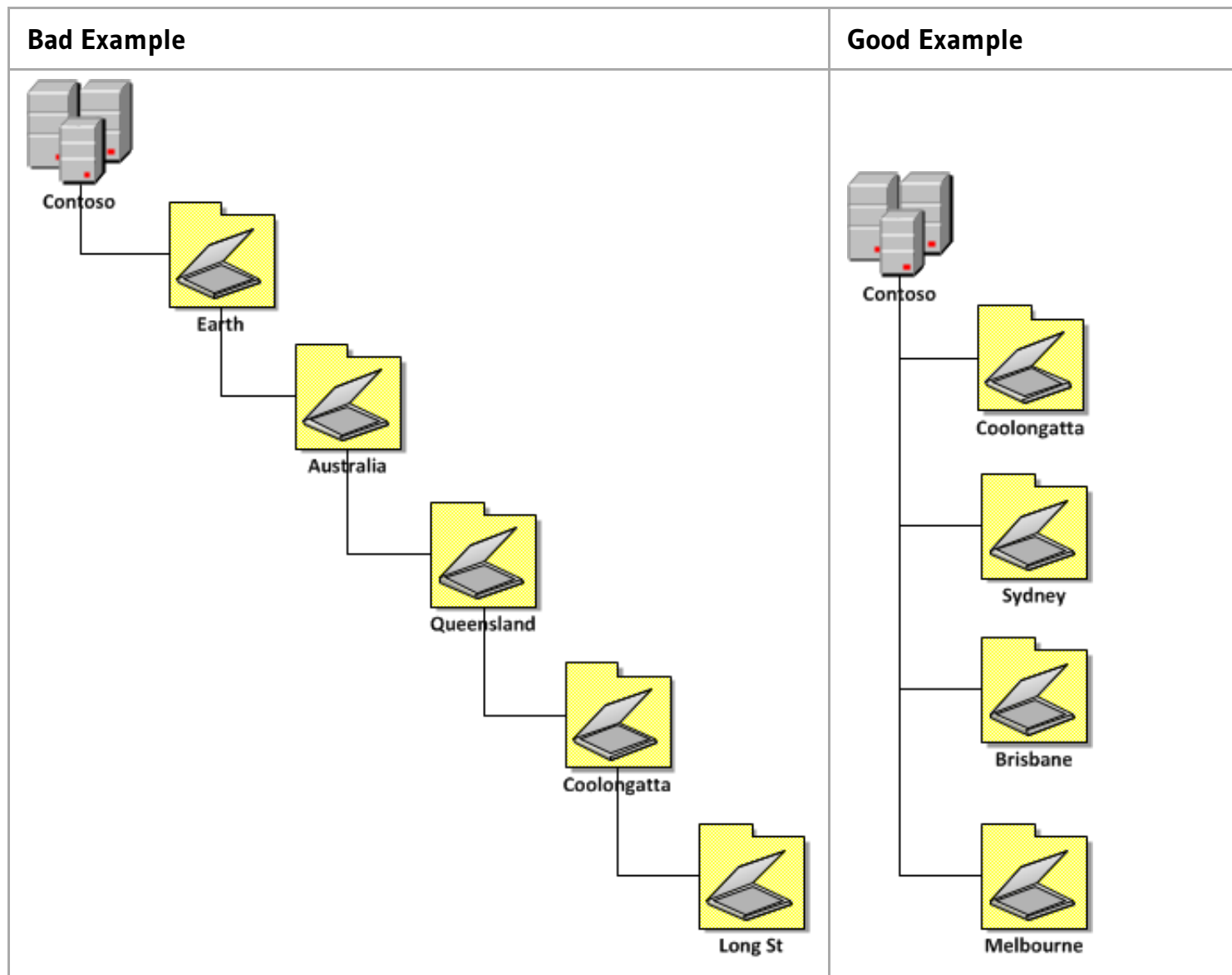| Bad Example	Good Example

Most to least significant from left to right

OU structures in AD are hierarchical therefore you need make your design fit to this structure. When deciding how your want to organise your OU structure you are probably going decide to make it either organisational or geographical. This is most important when you are going to a Geographical design as it is a physical impossibility to have one location located in two difference cities,states,countries or regions.

Bad Example	Good Example

Go wide not deep

As a general rule you should only start creating another OU level if you are actually going to do something to that OU (e.g. delegate security or apply a group policy). Don't be tempted to create an elaborate structure to organise you AD object if there is not reason to do so. Having a deep OU structure also makes it very difficult to delegate security in the same delegating security on multiple folders deep to folder on a file share.

Bad Example	Good Example

Be consistent

Don't mix your terms when naming our OU Structure as this leads to confusion if for IT admins that leads them to believe that something might be different about the two OU's where they actually contain the same type of objects. The example below shows how two different sites calls the OU for the computer in the organisation Workstations and Desktops.

Bad Example	Good Example