OSS Naming Convention Samples

Version 1.0

Contents

1. Introduction

The purpose of this book is to provide guidance on the considerations you will need to make when creating an OSS Naming Convention for your organisations. There are an almost limitless set of options for you to consider so this book helps narrow down the alternatives.

1.1. Overarching Principles

Each Operations Support System (OSS) relies on standard names for the data in its database. These standard formats are usually called "Naming Conventions." These are important because they make sure that all users create data that is consistent in the OSS database.

The OSS introduces a new level of naming convention. The naming conventions discussed in this document only refer to the naming of the objects in the OSS database level rather than NE/EMS/NMS level naming. This is described in more detail in Figure 1 and Figure 2 below.

Using the concept of alternate names it is possible that the "EMS name" of an object (eg a network element) can actually different to the "OSS name" in the OSS database. This is only applicable if alternate name functionality is supported by a vendor's OSS tools. Most tools do support this functionality.

Your organisation may need to make some changes to the "EMS Names" in the various NMS (Network Management Systems), EMS (Element Management Systems) or directly on the elements to ensure that it is compatible with the OSS and consistent in the OSS. One example will be to ensure that all Equipment Names are globally unique throughout the network, otherwise the OSS will not be able to differentiate between any network elements with the same name. Previously, each NMS or EMS only required locally unique names, but now that all NMS and EMS are consolidated under the OSS, global uniqueness is essential.

1.2. Naming Convention Levels

As indicated above, this document only refers to the OSS Level naming conventions. The following two diagrams describe this in more detail.

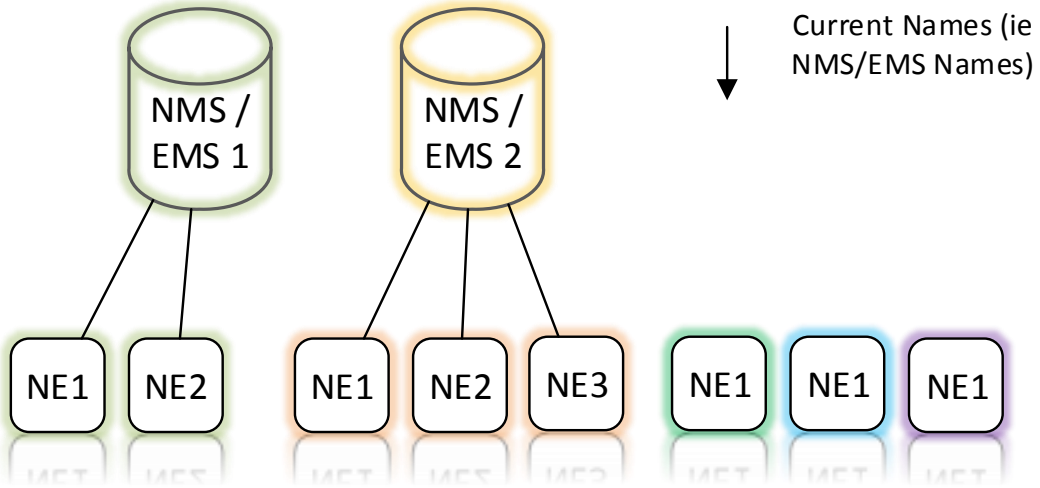

Figure 1 Before Implementing the OSS

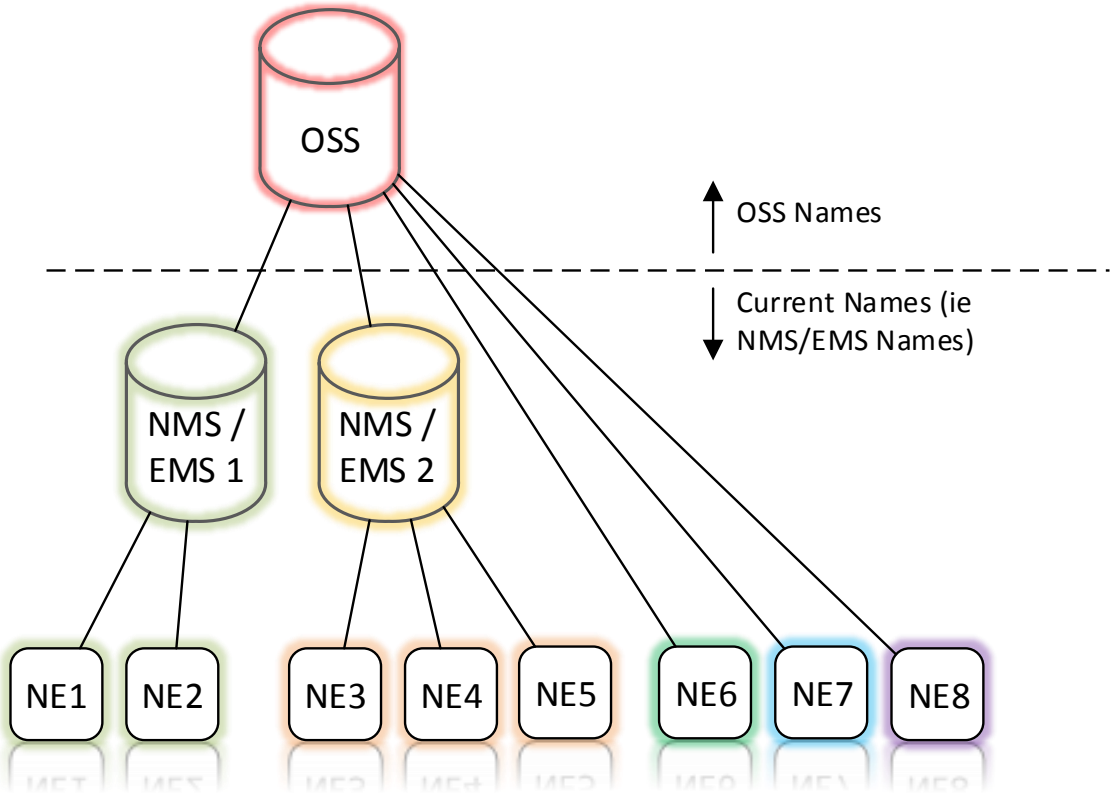

Figure 2 After Implementing the OSS

There are some objects, such as Equipment name, that must be globally unique after the OSS is in place. Similarly, the operators of the OSS must be able to easily determine what type of equipment is used.

In the current situation (before OSS), "Equipment Name" (for example) only needs to be unique within the realm of the EMS that controls it. If we use the NE's shown above, it is currently possible for the EMS1/NE1 and EMS2/NE1 to both have a device called NE1 without causing conflicts. However, after the OSS is added, every network element must have a unique name, so one of these devices must be changed from NE1 to something else that is globally unique.

Similarly, the OSS operators are now responsible for all of the equipment in figure 2 so it may sometimes be important for the naming convention to contain a greater level of detail. Before the OSS, it was suitable for the operators of EMS1 to call a device HNI01 and for EMS3 to call a device HCM01 because the operators automatically knew which type of equipment these related to (eg a PABX).

However, after implementation the OSS operators will manage devices under all EMS, so an equipment name like HNI01 may not contain enough information because it doesn't identify what type of device it is. It would be better for the OSS operators to have an "OSS naming convention" that shows the type of device they are looking at (eg HNI-PABX-01 and HCM-MD110-01).

If your Organisation wants to have an OSS name that is in the format HNI-PABX-01, but also keep the current EMS name HNI01, this is possible because the OSS software should have a field that allows the Organisation to store an "OSS name" and an "EMS name" that are different. Ideally the "OSS

name" and "EMS name" would be equal, so it would be better if the EMS name could also be changed to HNI-EWSD-01. However, this means manually changing the names in many devices and can take an inordinate amount of time.

1.3. Choosing a Naming Convention

When choosing a suitable set of naming conventions under an OSS domain, the following must be taken into consideration:

- The first step is to identify the objects that require names in the OSS database. This can range from network assets such as network elements, circuits, rings, etc through to customer codes, network operating regions and a vast range of other objects
- There is no single best approach to naming conventions
- Naming conventions should be chosen that are most meaningful to the operational business units that rely on the nomenclature. If this means retaining legacy naming conventions that have been in use for many years, then the new naming conventions should attempt to mimic these as closely as possible
- Developing OSS-level naming conventions often have to take into account the legacy characteristics of the NMS-level naming conventions that reside in the existing network management or documentation
- Naming convention character lengths are a trade-off between the breadth of data they represent (more characters) and the ease of use (less characters)
- In addition to operation of the OSS, naming conventions must take into account operational issues such as physical labelling of assets, the implication of data updates (eg changing the naming convention may result in changing configurations in hundreds of devices or thousands of network drawings), etc
- We expect that it will be difficult to change naming rules because it will initiate changes that could be made down to NMS/EMS/NE level in the network. This can represent a significant amount of work, particularly if the devices aren't remotely accessible (ie a field visit is required).
- However, we should remember that the changes won't need to be propagated down to the network if we are able to introduce "alternate names" in the OSS
- In the following sections, we propose nomenclature that has a structured number of digits or characters (eg site code has 4 characters). Whilst it is possible to have nomenclature of variable character length, it can make coding of ad-hoc reports or SQL queries more difficult, so it is recommended to have fixed lengths where possible. Where it is not possible to have fixed length fields, it may be possible to have separators like the "-" character to simplify SQL queries
- Each vendor's applications have their own unique characteristics with regards to naming conventions. The vendor's experts should be consulted when refining naming conventions to suit.

1.4. Naming Convention Standards

There is no common standard for naming conventions but we base the following naming conventions examples upon one of ITU-T's approaches to naming.

ITU-T M.1400 is a good basis from which to design naming conventions. M.1400 shows a number of differing recommendations, but the table below shows the basic format of the circuit names.

TABLE 1/M.1400

Format of designation	Town A	/	Suffix	–	Town B	/	Suffix		Function code	Serial number	
Signs	Characters	Slash	Letters/ digits	Hyphen	Characters	Slash	Letters/ digits	Space	Letters/ digits	Digits	
Number of characters	≤ 12	1	≤ 3	1	≤ 12	1	≤ 3	1	≤ 6	≤ 4	
									↑ No space		
Suggestion ->	BuildingA				BuildingB				Function	Index	Suffix

Because there are a number of different recommendations in M.1400, we prefer to simplify it a little, making 4 blocks (red, blue, green and purple) instead of the six shown by ITU above. We also prefer to add one extra block (yellow) onto the end for special cases (perhaps such as T for Test circuits, C7 for signalling circuits, or even EC if the circuit has echo cancellation [see section 2.8 of M.1400]).

That is, a circuit name should be in the format:

BLDGA-BLDGB-FUNCTION-INDEX-SUFFIX

So the following applies:

- Block 1 is red and usually identifies BuildingA (in some cases EquipmentA)
- Block 2 is blue and usually identifies BuildingB (in some cases EquipmentB)
- Block 3 is green and identifies the Function (or perhaps the speed) of a circuit. This should identify the type of "circuit." Examples are shown in sections 1.2.1, 3.2.1, 5.2.1, 12.1(b), 13.1(b), 14.1(b), etc of ITU-T Recommendation M.1400.
- Block 4 is purple and identifies the index of a circuit. It ensures that the name is unique if there are more than one circuit of the same type between BuildingA and BuildingB
- Block 5 is yellow (or Suffix) and can be used for special cases. It's not directly part of ITU recommendations, although the Index part can be used to store Index + Suffix.

It should be noted that sometimes EquipmentA/B notation is used, but this usually means that a long circuit name will occur. So in almost all cases, the carrier chooses LocationsA/B, which is consistent with what M.1400 suggests. EquipmentA/B is not really consistent with M.1400.

1.5. Additional Notes

One other important item to note is the characters that are used. There is a small problem with M.1400 because it recommends the use of the slash "/" character. This is a special character for some operating systems or applications which can cause problems that can be very difficult to isolate

So to avoid any possible problems we suggest that to use conventions with the following characters:

> **OK** : a-z, A-Z, 0-9, '-', '_'

But not the following characters

> &, @, ?, **/**, \, !, %, ', ", *, accentuated characters like é,è,ù, double-byte characters (eg Chinese, Vietnamese, etc).

2. Naming Convention Examples

The following sections provide some examples of choosing a naming convention. Your organisation will have your own different variations and preferred structures but hopefully it provides a starting point.

2.1.1. Location ID

The recommendation for LocationID is:

LocationID	
<Site><Region><BuildingIndex>	AAAABcc

Where

<Site> = AAAA [4 chars] (eg ABFT, ACKR, etc.).

<Region> = B [1 char] (ie **N**orthern, **C**entral, or **S**outhern)

<BuildingIndex> = cc [1 or maybe 2 digit sequential number indicating the building number at the site].

Example

ABFTS01

Notes:

An alternative for LocationID is to simply use the "Site" code of 4 characters if the Region and/or BuildingIndex are of relatively minor significance operationally. If we use the same example above, the LocationID would simply be ABFT

If none of the sites have more than 10 buildings this would imply that a single digit for the BuildingIndex would be sufficient to cater for the current situation, but it doesn't allow for future possible expansion. As such, 2 digits have been suggested.

If it causes operational confusion, we can change BuildingIndex to alphanumeric rather than integer.

LocationID could be extended to incorporate Room ID as well, but it is not recommended because we would prefer to keep LocationID as short as possible so as to keep the Circuit ID shorter and more manageable.

2.1.2. Equipment ID

The recommendation for EquipmentID is:

EquipmentID	
<LocationID> - <TNE> - <Index>	AAAABcc-DDDD-eee

Where

<LocationID> = AAAABcc [7 chars] (see section 2.1.1).

<TNE> = DDDD [from 3 to 8 characters]

TNE is short for “Type of Network Equipment”

<Index> = eee [3 digit sequential number]

Example

ABFTS01-ADM16-001

Notes:

There are a number of possible approaches for defining the Type of Network Equipment. TNE could define the device’s functionality. For example an SDH MUX with STM16 line rate could be designated ADM16 regardless of which make/model is used. Alternatively, a convention that aligns with make and model might be more operationally useful, such as OSN1500 (ie the Huawei OptiX OSN 1500). In the IP context, a router could be designated by its function (core, distribution, CE, PE) or by make / model.

2.1.3. Circuit ID

The recommendation for CircuitID is:

CircuitID	
<LocationID[1]> - <LocationID[2]> <Speed> <Index> {Suffix}	AAAABcc[1]- AAAABcc[2] FFFF gggg {HHH}

Where

<LocationID> = AAAABcc [7 chars] (see section 2.1.1).

Note
LocationID[1] is the building at the A-end of the circuit and
LocationID[2] is the building at the B-end of the circuit

<Speed> = FFFF [from 2 to 5 characters] (see "Speeds" in the appendix)

<Index> = gggg [4 digit sequential number]

{Suffix} = HHH [3 chars] (not mandatory)
This is generally used for special purposes, such as priority, signalling, test, protection, etc.

Example

ABFTS01-ACKN001 E1 0001 MUX

Notes:

- Remember that we don't need to record all supplementary information in the circuit name. It will be stored in other fields in the database, but we will be able to quickly see the supplementary features of each circuit using the OSS software.

2.1.4. Ring ID

The recommendation for RingID is:

RingID	
<Region><RingIndex>	Biii

Where

<Region> = B [1 char] (ie **N**orthern, **C**entral, or **S**outhern)

<RingIndex> = iii [2 or maybe 3 digit sequential number indicating the ring number within the region].

Example

N-001

Notes:

- Not all NEMS use the Ring ID object so this could be removed
- There are many different alternatives for ring names and no known international standard
- Using the region as an identifier may not suit the Organisation's network if there are rings that cross regions or are not easily identified as being primarily within a given region. As such, an alternative nomenclature may need to be suggested
- If regions aren't relevant then something as simple as Ring<iii> (eg Ring-001) could be used.
- A suffix could be added if there are certain functionalities that are required within a ring, such as servicing a particular customer or function. The suffix could also represent the method of protection within the ring (eg 2F MSSpring)
- Some transmission NMS require the ring nomenclature, whilst others don't, so that may have an impact on whether we need a naming convention for it or not.

2.1.5. Container ID

The recommendation for ContainerID is:

ContainerID	
<LocationID> - <ContainerType> - <Index>	AAAABcc-JJJ-kkk

Where

<LocationID> = AAAABcc [7 chars] (see section 2.1.1).

<ContainerType> = JJJ [from 3 to 8 characters] (see "Types of Container" in the appendix)

<Index> = kkk [3 digit sequential number]

Example

ABFTS01-RAC-001

Notes:

There are a number of possible approaches for defining the Container Type. This could define the device's functionality. For example a Hallam 19" rack could be designated RAC regardless of which make/model or rack height is used. Alternatively, a convention that aligns with make and model or even the number of Rack Units (RU) might be more operationally useful

Other Container Types might be:

- Patch Panels (eg FTP – Fibre Termination Panel)
- MDF (Main Distribution Panel)
- Road-side Cabinets
- Manholes / Pits
- Joint Enclosures
- Etc

As with circuits, there may be a hierarchy of objects, such as multiple patch panels being installed in a 19" rack as well as being able to house various other types of equipment, power / batteries, air-conditioning, etc that need to be facilitated within the OSS applications and naming.

