

# Entender a propriedade do serviço NSO

## Contents

---

[Introdução](#)

[Pré-requisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Conceitos básicos](#)

[Propriedade do dispositivo](#)

[Propriedade em ação](#)

[Demonstração: Somente serviço possuído](#)

[Demonstração: somente dispositivo possuído](#)

[Demonstração: de propriedade do dispositivo + de propriedade do serviço](#)

[Reconciliar](#)

[Propósito](#)

[Transferir propriedade para serviços](#)

[Remover sinalizadores de valor originais](#)

[Corrigindo a propriedade do serviço](#)

[Reconciliar como reimplantação](#)

[Reconciliar Trabalhos Internos](#)

[Problemas envolvendo a propriedade do serviço](#)

[Falha na reconciliação](#)

[O serviço está excluindo a configuração que não é de propriedade do serviço](#)

[Demonstração: o serviço está excluindo a configuração que não é de propriedade do serviço](#)

[Keep-non-service-config vs Discard-non-service-config](#)

[Não Criar](#)

[Mesclar](#)

---

## Introdução

Este documento descreve o conceito geral, armadilhas comuns e soluções de propriedade de serviço no Cisco® Network Service Orchestrator (NSO).

## Pré-requisitos

### Requisitos

Este documento aplica-se a todas as versões de NSO atualmente disponíveis, incluindo NSO 6. O comportamento descrito só é aplicável a configurações NSO usando uma combinação de serviços e configuração não serviço. Embora os comandos específicos usados nos exemplos deste documento se apliquem somente ao NED (Network Element Driver) usado, a lógica

subjacente se aplica a qualquer dispositivo gerenciado pelo NSO.

## Componentes Utilizados

- NSO 6.1.6
- NED: test-end 1.0 , um NED netconf personalizado que usa esse modelo yang.

```
NED yang model:
module test-ned{
  namespace "http://example.org/ned/service-ownership";
  prefix ownership;
  import ietf-inet-types{ prefix inet;}

  list interface {
    key interface-name;
    leaf interface-name{
      type string;
    }

    leaf ip-address {
      type inet:ipv4-address;
    }

    leaf description {
      type string;
    }
  }
}
```

- Serviço: exemplo-serviço 1.0: um serviço personalizado construído usando este modelo e modelo yang

```
module example-service {
  namespace "http://com/example/exampleservice";
  prefix example-service;

  import ietf-inet-types {
    prefix inet;
  }
  import tailf-ncs {
    prefix ncs;
  }

  list example-service {
    key name;

    uses ncs:service-data;
    ncs:servicepoint "example-service";

    leaf name {
      type string;
    }
    leaf-list device {
```

```
    type leafref {
      path "/ncs:devices/ncs:device/ncs:name";
    }
  }
  leaf ipaddress {
    type inet:ipv4-address;
  }
}
}
```

{/device}

FE1

{/ipaddress}

As informações neste documento foram criadas a partir de dispositivos em um ambiente de laboratório específico. Todos os dispositivos utilizados neste documento foram iniciados com uma configuração (padrão) inicial. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

## Conceitos básicos

A finalidade da propriedade do serviço é permitir que o NSO rastreie qual configuração está relacionada a qual serviço. Quando você exclui um serviço, o NSO precisa excluir a configuração relacionada a ele e ele usa a propriedade do serviço para determinar qual configuração excluir. Quando a configuração é de propriedade de mais de um serviço, a exclusão de um dos serviços simplesmente remove essa referência de propriedade; A configuração em si permanece no banco de dados NSO (CDB) e nos dispositivos da rede.

A propriedade é exibida por meio de recontagens e ponteiros traseiros. Uma refcount mostra quantas entidades possuem essa parte da configuração. Um refcount é igual à quantidade de instâncias de serviço mais 1 se a configuração também for de propriedade do dispositivo. Um backpointer mostra o caminho dessas instâncias de serviço. Não há backpointer para exibir "device owned" (propriedade de dispositivo). Os ponteiros dos fundos só são mostrados no CDB para listas e contentores. Folhas individuais não mostram seus ponteiros traseiros, mas herdam de seus pais.

## Propriedade do dispositivo

Além de a configuração ser de propriedade de um serviço, ela também pode ser de propriedade do dispositivo. Às vezes, isso é chamado de "propriedade do dispositivo" ou "não pertencente ao serviço". Embora este documento use "de propriedade do dispositivo", observe que, embora isso seja mais fácil de entender, a propriedade que não seja de serviço não precisa incluir dispositivos. As configurações de LSA ou serviços empilhados podem ter uma configuração que não seja de propriedade do serviço, sem dispositivos.

A configuração é de propriedade do dispositivo quando a configuração é adicionada ao CDB sem o uso de uma implantação de serviço, mas em vez disso usa um método como sync-from, load merge ou ncs\_cli para definir a configuração. Quando uma instância de serviço assume a propriedade de uma configuração que já pertencia ao dispositivo, o refcount é definido como 2 para refletir a propriedade compartilhada. Quando a instância de serviço é excluída, a configuração não é excluída, apesar de apenas uma instância de serviço possuir a configuração antes da exclusão. Além disso, a configuração de propriedade do dispositivo adiciona uma marca de "valor original". Se uma instância de serviço substituir a configuração de propriedade do dispositivo e o serviço for excluído posteriormente, a configuração será restaurada para o valor original.

A propriedade do dispositivo é atribuída somente se a configuração ainda não existir no CDB quando você adicioná-la por meios que não sejam de serviço. A configuração de propriedade do serviço não se torna propriedade do dispositivo após sync-from. Mas a configuração de propriedade do dispositivo se torna de propriedade do dispositivo e de propriedade do serviço se você implantar um serviço no topo.

## Propriedade em ação

### Demonstração: Somente serviço possuído

Quando você implanta um serviço enquanto a configuração de destino está vazia, o serviço cria a configuração e toma posse dela. O usuário pode verificar a propriedade usando um comando `show running-config` e anexar `| exhibit` metadados de serviço. Embora não seja obrigatório, recomenda-se também acrescentar `| exhibit xml` como a saída de estilo CLI padrão nem sempre reflete corretamente como os dados são modelados no CDB.

```
admin@ncs(config)# do show running-config devices device mydevice0 config
% No entries found.
admin@ncs(config)# example-service s1 device mydevice0 ipaddress 192.0.2.1
admin@ncs(config-example-service-s1)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          +       interface FE1 {
          +           ip-address 192.0.2.1;
          +       }
        }
      }
    }
  }
  +example-service s1 {
  +  device [ mydevice0 ];
  +  ipaddress 192.0.2.1;
  +}
}
admin@ncs(config-example-service-s1)# commit
Commit complete.
admin@ncs(config-example-service-s1)# do show running-config devices device mydevice0 config | display
```

mydevice0

FE1

192.0.2.1

Além disso, se você adicionar uma segunda instância de serviço direcionada para a mesma configuração, a propriedade será compartilhada pelas duas instâncias de serviço. O Refcount é 2 e há 2 backpointers.

```
admin@ncs(config-example-service-s1)# example-service s2 device mydevice0 ipaddress 192.0.2.2
admin@ncs(config-example-service-s2)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          interface FE1 {
            - ip-address 192.0.2.1;
            + ip-address 192.0.2.2;
```

```
        }
    }
}
+example-service s2 {
+  device [ mydevice0 ];
+  ipaddress 192.0.2.2;
+}
}
}
admin@ncs(config-example-service-s2)# commit
Commit complete.
admin@ncs(config-example-service-s2)# do show running-config devices device mydevice0 config | display
```

mydevice0

FE1

192.0.2.2

## Demonstração: somente dispositivo possuído

Quando você adiciona dados ao CDB usando a combinação de carga, ncs\_cli ou sync-from, sem o uso de um serviço, esses dados se tornam de propriedade do dispositivo. O refcount e o backpointer estão ocultos.

```
admin@ncs(config)# no example-service
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)# do show running-config devices device mydevice0 config
% No entries found.
admin@ncs(config)# load merge merge-config.xml
Loading.
386 bytes parsed in 0.00 sec (137.22 KiB/sec)
admin@ncs(config)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          +       interface FE1 {
          +         ip-address 192.0.2.1;
          +       }
        }
      }
    }
  }
}
admin@ncs(config)# commit
Commit complete.
admin@ncs(config)# do show running-config devices device mydevice0 config | display service-meta-data |
```

FE1

192.0.2.1

## Demonstração: de propriedade do dispositivo + de propriedade do serviço

Este exemplo demonstra como criar facilmente dispositivos combinados e propriedade de serviço usando um serviço e sync-from.

Você implanta o serviço e, em seguida, exclui o serviço somente no CDB usando Commit no-networking. Dessa forma, a configuração ainda existe no dispositivo final. Quando você executa a sincronização a partir de, a configuração é adicionada de volta no CDB, mas é de propriedade do dispositivo em vez de de propriedade do serviço. Lembre-se de que show running-config quando executado no NSO mostra os dados do CDB, não os dados atualmente nos dispositivos.

```
admin@ncs(config)# no devices device mydevice0 config
admin@ncs(config)# commit
admin@ncs(config)# do show running-config devices device mydevice0 config
% No entries found.
```

```
admin@ncs(config)# example-service s1 device mydevice0 ipaddress 192.0.2.1
admin@ncs(config-example-service-s1)# commit
admin@ncs(config-example-service-s1)# top
admin@ncs(config)# no example-service s1
admin@ncs(config)# commit no-networking
Commit complete.
admin@ncs(config)# devices device mydevice0 sync-from
result true
admin@ncs(config)# do show running-config devices device mydevice0 config | display service-meta-data |
```

mydevice0

FE1

192.0.2.1

Após sync-from, a configuração só pertence ao dispositivo. Refcounter está oculto. Ao implantar o serviço novamente, o refcount se torna 2, mas o backpointer exibe apenas uma única instância de serviço. O segundo refcounter representa a propriedade do dispositivo. As mesmas regras se aplicam como com a propriedade de serviço compartilhado, se o serviço for excluído, a configuração não será removida porque o dispositivo também possui parcialmente a configuração. Além disso, se os dados de serviço não corresponderem ao "valor original" como armazenado nos metadados de serviço, o NSO reverterá o valor de volta ao "valor original" se o serviço for removido.

```
admin@ncs(config)# example-service s1 device mydevice0 ipaddress 192.0.2.2
admin@ncs(config-example-service-s1)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          interface FE1 {
            - ip-address 192.0.2.1;
            + ip-address 192.0.2.2;
          }
        }
      }
    }
  }
  +example-service s1 {
  + device [ mydevice0 ];
  + ipaddress 192.0.2.2;
  +}
}
}
admin@ncs(config-example-service-s1)# commit
Commit complete.
admin@ncs(config-example-service-s1)# do show running-config devices device mydevice0 config | display
```

mydevice0

FE1

192.0.2.2

## Reconciliar

Sintaxe: <path-to-service instance> re-deploy reconcile

Sinalizadores opcionais: { keep-non-service-config } dry-run { outformat native }

### Propósito

O objetivo principal da funcionalidade de reconciliação é permitir que os usuários se livrem da propriedade indesejada de dispositivos e transfiram a propriedade totalmente para os serviços. Quando os usuários já têm uma rede em funcionamento e estão tentando transferir a propriedade para o NSO, eles normalmente apresentam primeiro a configuração por meio de operações de sincronização. Quando o CDB tiver toda a configuração de rede, o usuário implementará instâncias de serviço sobre a configuração existente. Neste ponto, a configuração ainda é de propriedade do dispositivo, o que limita a capacidade do serviço de excluir a configuração. Quando os usuários querem dar a seus serviços a propriedade total da configuração, eles podem

usar a funcionalidade de reconciliação, que faz 3 coisas.

- 1) 1) Transferir a propriedade para serviços
- 2) 2) Remover as opções de "valor original"
- 3) 3) Correção da propriedade do serviço

#### Transferir propriedade para serviços

A opção Reconciliar avalia toda a configuração de propriedade de um serviço e, se encontrar qualquer configuração de propriedade desse serviço e do dispositivo ou de outro modo não pertencente ao serviço, ela removerá a propriedade desse dispositivo, tornando o serviço o proprietário exclusivo. Reduzindo o refcount em 1.

Observação: se dois serviços tiverem uma parte da configuração e essa configuração também não for de propriedade do serviço, o refcount será 3. A reconciliação de qualquer um dos serviços remove essa propriedade de não-serviço, reduzindo o refcount a 2 para refletir ambos os serviços.

#### Remover sinalizadores de valor originais

Quando um serviço é implantado e sobregrava dados que não são de propriedade do serviço, refcount torna-se 2 e NSO adiciona uma marca de "valor original". Se a instância de serviço for excluída, o NSO tentará reverter para o valor original que existia antes do serviço.

Durante a reconciliação, não só o refcount é reduzido, mas o valor original é removido. A exclusão do serviço agora torna esse valor vazio ou o altera para um valor padrão conforme definido pelo modelo yang.

#### Corrigindo a propriedade do serviço

Em algumas situações, a propriedade pode ser atribuída incorretamente. A configuração de propriedade do dispositivo pertence incorretamente ao serviço ou a configuração pertence incorretamente ao serviço e ao dispositivo, embora se espere que pertença apenas ao serviço. Reconciliar pode corrigir esses desalinhamentos. Isso é importante para evitar problemas em que o serviço exclui a configuração que não pertence ao serviço.

#### Reconciliar como reimplantação

Reconciliar é uma subfunção da reimplantação do serviço. Se um serviço não estiver em sincronia com o CDB, a operação de reconciliação também executa uma redistribuição além de executar a função de reconciliação.

#### Reconciliar Trabalhos Internos

Embora os detalhes exatos de como a reconciliação opera sejam conhecidos apenas pelos

desenvolvedores, este documento fornece uma compreensão simplificada:

- 1) NSO corrige a propriedade do serviço
- 2) O NSO exclui do CDB toda a configuração de propriedade desta instância de serviço, mesmo que também seja de propriedade de outros serviços ou de propriedade do dispositivo
- 3) O NSO reimplanta a instância de serviço
- 4) O NSO restaura recontagens de serviço e ponteiros de fundo de outros serviços

## Problemas envolvendo a propriedade do serviço

### Falha na reconciliação

Caso você encontre trabalhos de "reimplantação", mas a "reconciliação de reimplantação" falhe: isso pode indicar que você encontrou um conflito entre o design do serviço e a forma como a função de reconciliação opera.

O problema surge do código de serviço que tenta ler a configuração do CDB que o serviço depois implanta. Você pode implantar este serviço somente porque a configuração já existe parcialmente no CDB antes da implantação. Mas durante a reconciliação, o NSO exclui temporariamente toda a configuração que pertence a esse serviço, incluindo a configuração que o serviço está tentando ler durante a reimplantação do serviço na próxima etapa. Isso geralmente resulta em um erro java ou python que relata a falha na leitura dos dados.

### O serviço está excluindo a configuração que não é de propriedade do serviço

Neste cenário, você encontra o NSO excluindo a configuração não pertencente ao serviço durante a exclusão ou reimplantação da instância de serviço. Isso ocorre porque a instância de serviço criou e possui a configuração original e, posteriormente, você adicionou manualmente (por meio de `ncs_cli`, `sync-from` ou outro método) uma parte da configuração dentro de um contêiner ou lista de propriedade de serviço.

Essa nova parte da configuração não deve pertencer ao serviço, mas como o serviço tem propriedade total do contêiner ou da lista, o serviço acaba por possuí-lo indiretamente.

Para resolver isso, use `redploy reconcile { keep-non-service-config }` para corrigir a propriedade do serviço. Ao fazer essa reconciliação, o `refcount` do contêiner ou lista aumenta para refletir se esse contêiner ou lista tem nós filhos de propriedade de serviço e não-propriedade de serviço. Dentro do nó pai, somente os nós de propriedade de serviço têm um `refcount` e um `backpointer`.

### Demonstração: o serviço está excluindo a configuração que não é de propriedade do serviço

Começando com uma única instância de serviço implantada com propriedade total, adicione manualmente uma descrição à interface usando `ncs_cli`.

```
admin@ncs(config-example-service-s1)# do show running-config devices device mydevice0 config | display
```

```
mydevice0
```

```
FE1
```

```
192.0.2.1
```

```
admin@ncs(config-example-service-s1)# top
admin@ncs(config)# devices device mydevice0 config interface FE1 description "This is an example descri
admin@ncs(config-interface-FE1)# commit
Commit complete.
```

```
admin@ncs(config-interface-FE1)# do show running-config devices device mydevice0 config | display servi
```

mydevice0

FE1

192.0.2.1

This is an example description

Observe como o refcount em <interface> permanece 1 mesmo que a configuração de propriedade do dispositivo tenha sido adicionada. Ao tentar excluir a instância de serviço, a descrição também é excluída, embora não deva fazer parte da instância de serviço. Para evitar isso, você pode executar o comando reconcile.

```
admin@ncs(config-interface-FE1)# top
admin@ncs(config)# no example-service s1
admin@ncs(config)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          -       interface FE1 {
          -         ip-address 192.0.2.1;
          -         description "This is an example description";
          -       }
        }
      }
    }
  }
  -example-service s1 {
  -  device [ mydevice0 ];
  -  ipaddress 192.0.2.1;
  -}
}
}
admin@ncs(config)# abort
admin@ncs# config
Entering configuration mode terminal
admin@ncs(config)# example-service s1 re-deploy reconcile { keep-non-service-config }
admin@ncs(config)# do show running-config devices device mydevice0 config | display service-meta-data |
```

mydevice0

FE1

192.0.2.1

This is an example description

Após a reconciliação, o refcount para a interface da lista aumentou para 2. Enquanto isso, refcount em leaf ip-address, permaneceu como 1. A entrada de lista "interface FE1" contém dados de propriedade do serviço e dados de não propriedade do serviço. Ao usar a reconciliação, o NSO reavalia a propriedade e atribui as recontagens de acordo. A exclusão agora destina-se apenas às áreas que são totalmente de propriedade da instância de serviço. A descrição ou a entrada da lista não é excluída.

```
admin@ncs(config)# no example-service s1
admin@ncs(config)# commit dry-run
cli {
  local-node {
```

```

data devices {
    device mydevice0 {
        config {
            interface FE1 {
-                ip-address 192.0.2.1;
            }
        }
    }
-example-service s1 {
-    device [ mydevice0 ];
-    ipaddress 192.0.2.1;
-}
}
}

```

## Keep-non-service-config vs Discard-non-service-config

Às vezes, os usuários não entendem bem o uso de discard-non-service-config.

No exemplo de reconciliação, foi usado "keep-non-service-config". Se o descarte for usado, ele ficará assim:

```

admin@ncs(config)# example-service s1 re-deploy reconcile { discard-non-service-config } dry-run
cli {
    local-node {
        data devices {
            device mydevice0 {
                config {
                    interface FE1 {
-                        description "This is an example description";
                    }
                }
            }
        }
    }
}
}

```

O padrão é "keep-non-service-config". Se nenhuma opção estiver definida, o padrão NSO é keep. O descarte é raramente usado, pois a maioria dos usuários prefere manter o que está em sua rede, mesmo que não seja gerenciado pelo NSO. Reconcile { discard-non-service-config } dry-run pode ser usado para descobrir quais pontos de dados existem no CDB que não fazem parte da configuração de serviço, mas seriam excluídos se o serviço fosse excluído ou reimplantado.

## Não Criar

Uma alternativa para usar a reconciliação de reimplantação para corrigir a propriedade de serviço quando misturada com dados não pertencentes ao serviço é evitar o conflito usando a marca ncreate.

Esta é uma marca que pode ser usada no modelo de serviço XML. A documentação diz "ncreate: a mesclagem afeta somente os itens de configuração que já existem no modelo. Ele nunca cria a configuração com essa marca. Ele modifica apenas as estruturas de configuração existentes."

O uso dessa tag tem um efeito colateral interessante: como o serviço não cria o nó, ele não se apropria dele.

Isso é comumente usado para evitar situações em que o NSO tenta excluir configurações que o dispositivo não permite que sejam excluídas.

Observe que essa marca é herdada por nós filhos, isso significa que se você adicionar a marca ncreate à interface, ela também se aplicará a todos os nós dentro dessa interface, a menos que você os marque com uma marca diferente, como mesclar"

Adicione uma marca ncreate ao modelo de serviço. Se a interface FE1 não existir, nada será configurado.

```
{/device}
```

FE1

```
{/ipaddress}
```

Recompile e recarregue o encapsulamento e depois teste.

```
admin@ncs(config)# do show running-config devices device mydevice0 config | display service-meta-data |
No entries found.
admin@ncs(config)# example-service s1 device mydevice0 ipaddress 192.0.2.1
admin@ncs(config-example-service-s1)# commit dry-run
cli {
  local-node {
    data +example-service s1 {
      +   device [ mydevice0 ];
      +   ipaddress 192.0.2.1;
      +}
    }
  }
}
```

Mesmo que os mesmos parâmetros tenham sido definidos como antes, a interface ou qualquer configuração subjacente não é criada na configuração do dispositivo.

## Mesclar

Adicione uma marca de mesclagem na configuração dentro da interface. Não adicione uma marca a "interface-name", pois essa é a chave da interface de lista. A chave deve sempre ter permissão para herdar o comportamento da lista. Recompile e recarregue o pacote.

{/device}

FE1

{/ipaddress}

Configure manualmente a interface FE1 antes de implantar o serviço.

```
admin@ncs(config)# no example-service
admin@ncs(config)# commit
admin@ncs(config)# do show running-config devices device mydevice0 config | display service-meta-data |
No entries found.
admin@ncs(config)# devices device mydevice0 config interface FE1
admin@ncs(config-interface-FE1)# commit dry-run
cli {
```

```

local-node {
  data devices {
    device mydevice0 {
      config {
        +       interface FE1 {
        +       }
      }
    }
  }
}
admin@ncs(config-interface-FE1)# commit
Commit complete.
admin@ncs(config-interface-FE1)# top
admin@ncs(config)# example-service s1 device mydevice0 ipaddress 192.0.2.1
admin@ncs(config-example-service-s1)# commit dry-run
cli {
  local-node {
    data devices {
      device mydevice0 {
        config {
          interface FE1 {
            +       ip-address 192.0.2.1;
          }
        }
      }
    }
    +example-service s1 {
    +  device [ mydevice0 ];
    +  ipaddress 192.0.2.1;
    +}
  }
}
admin@ncs(config-example-service-s1)# commit
Commit complete.
admin@ncs(config-example-service-s1)# do show running-config devices device mydevice0 config | display

```

mydevice0

FE1

192.0.2.1

A interface tem um refcount oculto 1: a interface foi implantada usando ncs\_cli, mas tem uma marca ncreate no pacote de serviço; o serviço não se apropriou. É de propriedade do dispositivo.

O principal tem refcount 1: pertence somente ao serviço

Se a instância de serviço for excluída, ela só excluirá o endereço IP, pois essa é a única parte que pertence inteiramente ao serviço.

## Sobre esta tradução

A Cisco traduziu este documento com a ajuda de tecnologias de tradução automática e humana para oferecer conteúdo de suporte aos seus usuários no seu próprio idioma, independentemente da localização.

Observe que mesmo a melhor tradução automática não será tão precisa quanto as realizadas por um tradutor profissional.

A Cisco Systems, Inc. não se responsabiliza pela precisão destas traduções e recomenda que o documento original em inglês ([link fornecido](#)) seja sempre consultado.