

# Verständnis der Service-Eigentumsrechte des NSO

## Inhalt

---

### [Einleitung](#)

### [Voraussetzungen](#)

[Anforderungen](#)

[Verwendete Komponenten](#)

### [Grundlagen](#)

### [Geräteunabhängig](#)

### [Eigentumsrechte in Aktion](#)

[Schaukasten: Nur Service-Eigentümer](#)

[Showcase: Nur im Besitz des Geräts](#)

[Schaukasten: Geräte- und Service-Eigentümer](#)

### [Aussöhnen](#)

#### [Zweck](#)

[Übertragung des Eigentums auf Services](#)

[Originalwertmarkierungen entfernen](#)

[Korrigieren des Serviceeigentums](#)

[Bei erneuter Bereitstellung abstimmen](#)

[Kompatibilität der internen Arbeitsabläufe](#)

### [Probleme im Zusammenhang mit dem Eigentum an Services](#)

[Abgleich fehlgeschlagen](#)

[Der Service löscht die Konfiguration, die nicht zum Service gehört.](#)

[Showcase: Service löscht nicht serviceeigene Konfiguration](#)

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

### [Keine Erstellung](#)

[Zusammenführen](#)

---

## Einleitung

In diesem Dokument werden das allgemeine Konzept, die gängigsten Probleme und die Lösungen im Zusammenhang mit der Servicebereitstellung in Cisco® Network Service Orchestrator (NSO) beschrieben.

## Voraussetzungen

### Anforderungen

Dieses Dokument gilt für alle aktuell verfügbaren NSO-Versionen einschließlich NSO 6. Das

beschriebene Verhalten gilt nur für NSO-Konfigurationen mit einer Kombination aus Services und Non-Service-Konfigurationen. Während die in den Beispielen in diesem Dokument verwendeten spezifischen Befehle nur für den verwendeten Netzwerkelementtreiber (Network Element Driver, NED) gelten, gilt die zugrunde liegende Logik für alle vom NSO verwalteten Geräte.

## Verwendete Komponenten

- NSO 6.1.6
- NED: test-end 1.0 , ein speziell entwickeltes NED für Netconf, das dieses Yang-Modell verwendet.

```
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;
    }
  }
}
```

- Service: example-service 1.0 : ein maßgeschneiderter Service mit diesem Yang-Modell und Vorlage

```
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}

Die Informationen in diesem Dokument beziehen sich auf Geräte in einer speziell eingerichteten Testumgebung. Alle Geräte, die in diesem Dokument benutzt wurden, begannen mit einer gelöschten (Nichterfüllungs) Konfiguration. Wenn Ihr Netzwerk in Betrieb ist, stellen Sie sicher, dass Sie die möglichen Auswirkungen aller Befehle kennen.

## Grundlagen

Der Zweck des Service-Eigentumsrechts besteht darin, dem NSO die Nachverfolgung der Konfiguration zu ermöglichen, die mit dem jeweiligen Service in Zusammenhang steht. Wenn Sie einen Service löschen, muss der NSO die zugehörige Konfiguration löschen. Anhand der Service-Eigentumsrechte legt der NSO fest, welche Konfiguration gelöscht werden soll. Wenn die Konfiguration mehreren Services gehört, wird beim Löschen eines Services dieser Verweis einfach entfernt. Die Konfiguration selbst verbleibt in der NSO-Datenbank (CDB) und auf den Geräten im Netzwerk.

Die Eigentumsverhältnisse werden durch Rückzählungen und Rückverweise angezeigt. Ein Bericht zeigt, wie viele Einheiten diesen Teil der Konfiguration besitzen. Ein Refcount entspricht der Anzahl der Service-Instanzen plus 1, wenn die Konfiguration ebenfalls im Besitz des Geräts ist. Ein Backpointer zeigt den Pfad dieser Service-Instanzen an. Es gibt keinen Rückzeiger zum Anzeigen von "Geräte besitzen". In CDB werden nur für Listen und Container Backpoints angezeigt. Einzelne Blätter zeigen ihre Backpointer nicht, aber sie erben von ihren Eltern.

## Geräteunabhängig

Die Konfiguration ist nicht nur Eigentum eines Services, sondern kann auch Eigentum des Geräts sein. Dies wird manchmal auch als "geräteunabhängig" oder "nicht serviceunabhängig" bezeichnet. In diesem Dokument wird zwar der Begriff "Geräte-Eigentum" verwendet. Beachten Sie jedoch, dass Geräte, die nicht Teil des Service sind, nicht berücksichtigt werden müssen, obwohl dies verständlicher ist. LSA-Setups oder Stack-Services können ohne Geräte im Besitz von anderen Geräten sein.

Die Konfiguration befindet sich im Gerätebesitz, wenn die Konfiguration CDB hinzugefügt wird, ohne dass eine Servicebereitstellung verwendet wird. Stattdessen wurde eine Methode wie "sync-from", "load merge" oder "ncs\_cli" zum Festlegen der Konfiguration verwendet. Wenn eine Serviceinstanz die Zuständigkeit für die Konfiguration übernimmt, die bereits im Besitz des Geräts war, wird der Refcount auf 2 gesetzt, um den gemeinsamen Besitz widerzuspiegeln. Wenn die Serviceinstanz gelöscht wird, wird die Konfiguration nicht gelöscht, obwohl vor dem Löschen nur

eine Serviceinstanz im Besitz der Konfiguration ist. Darüber hinaus wird durch die geräteeigene Konfiguration ein "ursprünglicher Wert"-Tag hinzugefügt. Wenn eine Dienstinstanz die geräteeigene Konfiguration überschreibt und der Dienst später gelöscht wird, wird die Konfiguration auf den ursprünglichen Wert wiederhergestellt.

Der Gerätebesitz wird nur dann zugewiesen, wenn die Konfiguration beim Hinzufügen mithilfe von nicht servicebezogenen Mitteln noch nicht im CDB vorhanden war. Die diensteigene Konfiguration bleibt nach der Synchronisierung von nicht im Besitz des Geräts. Die geräteeigene Konfiguration steht jedoch sowohl im Eigentum der Geräte als auch der Services, wenn Sie einen Service als Ergänzung bereitstellen.

## Eigentumsrechte in Aktion

### Schaukasten: Nur Service-Eigentümer

Wenn Sie einen Dienst bereitstellen, während die Zielkonfiguration leer ist, erstellt der Dienst die Konfiguration und übernimmt die Verantwortung für diese Konfiguration. Der Benutzer kann den Besitzer überprüfen, indem er den Befehl `show running-config` verwendet und das Append | Service-Meta-Daten anzeigen. Obwohl nicht obligatorisch, wird empfohlen, auch anzufügen | zeigt xml an, da die Ausgabe im Standard-CLI-Stil nicht immer korrekt wiedergibt, wie die Daten in CDB modelliert werden.

```
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

Wenn Sie außerdem eine zweite Service-Instanz hinzufügen, die dieselbe Konfiguration anstrebt, wird der Besitz von den beiden Service-Instanzen gemeinsam genutzt. Der Refcount ist 2 und es gibt 2 Backpointer.

```
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

## Showcase: Nur im Besitz des Geräts

Wenn Sie CDB Daten mithilfe von Load Merge, ncs\_cli oder sync-from hinzufügen, ohne dass ein Dienst verwendet wird, sind diese Daten Eigentum des Geräts. Der Refcount und der Backpointer sind ausgeblendet.

```
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 |
```



mydevice0

FE1

192.0.2.1

## Schaukasten: Geräte- und Service-Eigentümer

In diesem Beispiel wird veranschaulicht, wie auf einfache Weise eine kombinierte Geräte- und Dienstereigentumsinformationen mithilfe eines Diensts und einer Synchronisierungsfunktion erstellt werden können.

Sie stellen den Service bereit und löschen ihn dann nur in CDB mit Commit no-networking. Auf diese Weise bleibt die Konfiguration auf dem Endgerät erhalten. Wenn Sie "sync-from" ausführen, wird die Konfiguration wieder in CDB hinzugefügt, aber sie ist im Besitz des Geräts und nicht des Services. Beachten Sie, dass show running-config bei Ausführung in NSO Ihre CDB-Daten

anzeigt, nicht die Daten, die sich derzeit auf den Geräten befinden.

```
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

Nach der Synchronisierung über ist die Konfiguration nur noch im Besitz des Geräts. Refcounter ist ausgeblendet. Wenn Sie den Dienst erneut bereitstellen, wird recount zu 2, aber der Backpointer zeigt nur eine einzelne Dienstinstanz an. Der zweite Refcounter stellt den Gerätebesitz dar. Es gelten die gleichen Regeln wie für den Besitz eines gemeinsam genutzten Diensts. Wenn der Dienst gelöscht wird, wird die Konfiguration nicht entfernt, da sich die Konfiguration teilweise im Besitz des Geräts befindet. Wenn die Servicedaten nicht mit dem ursprünglichen Wert übereinstimmen, der in den Service-Metadaten gespeichert ist, setzt der NSO den Wert auf den ursprünglichen Wert zurück, wenn der Service entfernt wird.

```
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

## Aussöhnen

Syntax: <Pfad-zu-Service-Instanz> redeploy reconcile

Optionale Flags: { keep-non-service-config } dry-run { outformat nativ }

### Zweck

Der Hauptzweck der Abstimmungsfunktion besteht darin, es Benutzern zu ermöglichen, unerwünschte Geräte zu entfernen und die Eigentumsrechte vollständig auf die Services zu

übertragen. Wenn Benutzer bereits über ein funktionierendes Netzwerk verfügen und versuchen, die Eigentumsrechte an den NSO zu übertragen, führen sie die Konfiguration in der Regel zuerst über Abgleich-Vorgänge ein. Sobald CDB über die gesamte Netzwerkkonfiguration verfügt, stellt der Benutzer zusätzlich zur vorhandenen Konfiguration Serviceinstanzen bereit. An diesem Punkt befindet sich die Konfiguration noch im Besitz des Geräts, wodurch die Möglichkeit des Service zum Löschen der Konfiguration eingeschränkt wird. Wenn Benutzer ihren Services die volle Verantwortung für die Konfiguration übertragen möchten, können sie die Abstimmungsfunktion verwenden, die drei Aufgaben übernimmt.

- 1) 1) Übertragung der Eigentumsrechte an den Services
- 2) 2) Entfernen Sie die Markierungen für "ursprüngliche Werte".
- 3) 3) Korrektur der Eigentumsverhältnisse bei den Dienstleistungen

### Übertragung des Eigentums auf Services

Reconcile evaluiert die gesamte Konfiguration eines Dienstes. Wenn eine Konfiguration gefunden wird, die sowohl diesem Dienst als auch dem Gerät gehört oder anderweitig nicht dem Dienst gehört, wird der Gerätebesitz entfernt, sodass der Dienst zum exklusiven Eigentümer wird. Reduzierung der Refcount um 1.

Hinweis: Wenn zwei Services eine Konfiguration besitzen und diese auch nicht vom Service verwaltet wird, lautet die Anzahl der Refcount-Anfragen 3. Durch den Abgleich eines der Services entfällt das Eigentum an diesen Nicht-Services, sodass die Anzahl der Neuzugänge auf 2 reduziert wird, um beide Services widerzuspiegeln.

### Originalwertmarkierungen entfernen

Wenn ein Service bereitgestellt wird und nicht servicebezogene Daten überschrieben werden, wird refcount zu 2, und der NSO fügt ein "Original Value"-Tag hinzu. Wenn die Serviceinstanz jemals gelöscht wird, versucht der NSO, auf den ursprünglichen Wert zurückzukehren, der vor dem Service bestand.

Während des Abgleichs wird nicht nur der Refcount reduziert, sondern der ursprüngliche Wert wird entfernt. Durch das Löschen des Service wird dieser Wert jetzt leer oder auf einen Standardwert gemäß der Definition des Yang-Modells geändert.

### Korrigieren des Serviceeigentums

In einigen Situationen kann die Zuweisung von Besitzrechten falsch sein. Entweder befindet sich die geräteeigene Konfiguration fälschlicherweise im Besitz des Diensts, oder die Konfiguration befindet sich fälschlicherweise im Besitz des Diensts und des Geräts, während davon ausgegangen wird, dass sie nur dem Dienst gehört. Reconcile kann diese Fehlausrichtung korrigieren. Dies ist wichtig, um zu vermeiden, dass der Dienst nicht diensteigene Konfigurationen löscht.

## Bei erneuter Bereitstellung abstimmen

Reconcile ist eine Unterfunktion der erneuten Dienstbereitstellung. Wenn ein Dienst nicht mit CDB synchronisiert ist, führt der Abstimmungsvorgang zusätzlich zur Ausführung der Abstimmungsfunktion auch eine erneute Bereitstellung aus.

## Kompatibilität der internen Arbeitsabläufe

Obwohl die genauen Details der Funktionsweise von reconcile nur den Entwicklern bekannt sind, bietet dieses Dokument ein vereinfachtes Verständnis:

- 1) NSO korrigiert Eigentumsrechte an Services
- 2) Der NSO löscht alle Konfigurationen dieser Serviceinstanz von CDB, selbst wenn diese ebenfalls anderen Services gehört oder im Besitz eines Geräts ist.
- 3) NSO stellt die Service-Instanz erneut bereit
- 4) Der NSO stellt Servicerefcounts und Backpointers (Rückverweise auf andere Services) wieder her

## Probleme im Zusammenhang mit dem Eigentum an Services

### Abgleich fehlgeschlagen

Falls "re-deploy" funktioniert, "re-deploy reconcile" jedoch fehlschlägt: Dies kann darauf hinweisen, dass ein Konflikt zwischen dem Dienstdesign und der Funktionsweise der Reconcile-Funktion aufgetreten ist.

Das Problem entsteht durch den Dienstcode, der versucht, die Konfiguration aus CDB zu lesen, die der Dienst dann später bereitstellt. Sie können diesen Dienst nur bereitstellen, weil die Konfiguration vor der Bereitstellung bereits teilweise in CDB vorhanden ist. Während des Abgleichs löscht der NSO vorübergehend alle Konfigurationen, die zu diesem Service gehören, einschließlich der Konfiguration, die der Service während der erneuten Bereitstellung des Service im nächsten Schritt zu lesen versucht. Dies führt in der Regel zu einem Java- oder Python-Fehler, der den Fehler beim Lesen der Daten meldet.

### Der Service löscht die Konfiguration, die nicht zum Service gehört.

In diesem Szenario löscht der NSO während des Löschens oder erneuten Bereitstellens der Serviceinstanz die Konfiguration, die nicht zum Service gehört. Dies liegt daran, dass die Dienstinstanz erstellt wurde und die ursprüngliche Konfiguration besitzt, und Sie später manuell (über `ncs_cli`, `sync-from` oder eine andere Methode) einen Teil der Konfiguration in einem diensteigenen Container oder einer diensteigenen Liste hinzugefügt haben.

Diese neue Konfiguration sollte nicht dem Dienst gehören, aber da der Dienst das vollständige Eigentum an dem Container oder der Liste besitzt, besitzt der Dienst sie indirekt.

Dies kann durch die Verwendung von `re-deploy reconcile { keep-non-service-config }` behoben werden, um das Eigentum an den Services zu korrigieren. Bei dieser Abstimmung wird der Refcount des Containers oder der Liste erhöht, um anzugeben, dass dieser Container oder diese Liste sowohl diensteigene als auch nicht diensteigene untergeordnete Knoten aufweist. Innerhalb des übergeordneten Knotens verfügen nur die diensteigenen Knoten über einen Refcount und einen Backpointer.

## Showcase: Service löscht nicht serviceeigene Konfiguration

Beginnen Sie mit einer einzelnen Dienstinstantz, die vollständig bereitgestellt wurde, und fügen Sie der Schnittstelle manuell eine Beschreibung mithilfe von `ncs_cli` hinzu.

```
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

Beachten Sie, dass der Zähler für <Schnittstelle> 1 bleibt, obwohl eine geräteeigene Konfiguration hinzugefügt wurde. Wenn Sie versuchen, die Dienstinstantz zu löschen, wird auch die Beschreibung gelöscht, obwohl sie nicht Teil der Dienstinstantz sein sollte. Um dies zu vermeiden, können Sie den Befehl reconcile ausführen.

```
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

Nach dem Abgleich wurde der Refcount für die Listenschnittstelle auf 2 erhöht. Inzwischen, refcount on leaf ip-address, has been 1. Der Listeneintrag "interface FE1" enthält sowohl diensteigene als auch nicht diensteigene Daten. Mithilfe von Reconcile evaluiert der NSO die Eigentumsrechte neu und ordnet Rückrechnungen entsprechend zu. Die Löschung betrifft nun nur die Bereiche, die vollständig der Serviceinstanz gehören. Weder die Beschreibung noch der Listeneintrag werden gelöscht.

```
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 und Discard-non-service-config

Benutzer missverstehen manchmal die Verwendung von "discard-non-service-config".

Im Reconcile-Beispiel wurde "keep-non-service-config" verwendet. Bei Verwendung von 'verwerfen' sieht es folgendermaßen aus:

```
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";
          }
        }
      }
    }
  }
}
}
```

Der Standardwert ist "keep-non-service-config". Wenn keine der Optionen definiert ist, behält der NSO die Standardeinstellung bei. Discard wird nur selten verwendet, da die meisten Benutzer es vorziehen, die Inhalte im Netzwerk zu behalten, selbst wenn diese nicht vom NSO verwaltet werden. Reconcile { discard-non-service-config } dry-run kann verwendet werden, um herauszufinden, welche Datenpunkte in CDB vorhanden sind, die nicht Teil der Dienstkonfiguration sind, aber gelöscht würden, wenn der Dienst gelöscht oder erneut bereitgestellt würde.

## Keine Erstellung

Eine Alternative zur Verwendung von Re-Deploy Reconcile zur Korrektur des Serviceeigentums in Verbindung mit nicht diensteigenen Daten besteht darin, den Konflikt mithilfe des ncreate-Tags zu verhindern.

Dies ist ein Tag, das in Ihrer XML-Dienstvorlage verwendet werden kann. In der Dokumentation heißt es "ncreate: Das Zusammenführen betrifft nur Konfigurationselemente, die bereits in der Vorlage vorhanden sind. Es wird niemals die Konfiguration mit diesem Tag erstellt. Es werden nur vorhandene Konfigurationsstrukturen geändert."

Die Verwendung dieses Tags hat einen interessanten Nebeneffekt: Da der Dienst den Knoten nicht erstellt, übernimmt er keine Verantwortung dafür.

Dies wird häufig verwendet, um Situationen zu vermeiden, in denen der NSO versucht, Konfigurationen zu löschen, die vom Gerät nicht gelöscht werden dürfen.

Beachten Sie, dass dieses Tag von untergeordneten Knoten geerbt wird. Dies bedeutet, dass es, wenn Sie das ncreate-Tag zur Schnittstelle hinzufügen, auch auf alle Knoten innerhalb dieser Schnittstelle angewendet wird, es sei denn, Sie markieren sie mit einem anderen Tag, z. B. "Zusammenführen".

Fügen Sie der Dienstvorlage ein ncreate-Tag hinzu. Wenn die Schnittstelle FE1 nicht vorhanden ist, wird keine Konfiguration vorgenommen.

```
{/device}
```

FE1

{/ipaddress}

Kompilieren und laden Sie das Paket neu, und testen Sie es anschließend.

```
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;
      +}
    }
  }
}
```

Obwohl die gleichen Parameter wie zuvor definiert wurden, wird die Schnittstelle oder eine zugrunde liegende Konfiguration in der Gerätekonfiguration nicht erstellt.

## Zusammenführen

Fügen Sie der Konfiguration innerhalb der Schnittstelle ein Merge-Tag hinzu. Fügen Sie "interface-name" kein Tag hinzu, da dies der Schlüssel der Listenschnittstelle ist. Der Schlüssel muss immer das Verhalten der Liste erben dürfen. Kompilieren und laden Sie das Paket neu.

```
{/device}
```

```
FE1
```

```
{/ipaddress}
```

Konfigurieren Sie die Schnittstelle FE1 manuell, bevor Sie den Dienst bereitstellen.

```
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

Schnittstelle hat einen verborgenen refcount 1 : Schnittstelle wurde mit ncs\_cli bereitgestellt, aber sie hat ein nocreate-Tag im Service-Paket; der Service hat nicht die Verantwortung übernommen. Es befindet sich im Besitz des Geräts.

Primary verfügt über refcount 1: It is only owned by the service



Wenn die Dienstinstanz gelöscht wird, würde sie nur ipaddress löschen, da dies der einzige Teil ist, der vollständig im Besitz des Diensts ist.

## Informationen zu dieser Übersetzung

Cisco hat dieses Dokument maschinell übersetzen und von einem menschlichen Übersetzer editieren und korrigieren lassen, um unseren Benutzern auf der ganzen Welt Support-Inhalte in ihrer eigenen Sprache zu bieten. Bitte beachten Sie, dass selbst die beste maschinelle Übersetzung nicht so genau ist wie eine von einem professionellen Übersetzer angefertigte. Cisco Systems, Inc. übernimmt keine Haftung für die Richtigkeit dieser Übersetzungen und empfiehlt, immer das englische Originaldokument (siehe bereitgestellter Link) heranzuziehen.