

Oracle® Solaris 11.2 でのネットワークコン ポーネントの構成と管理

ORACLE®

Part No: E53786
2014 年 7 月

Copyright © 2011, 2014, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクル社までご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアもしくはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアもしくはハードウェアは、危険が伴うアプリケーション（人的傷害を発生させる可能性があるアプリケーションを含む）への用途を目的として開発されていません。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用する場合、安全に使用するために、適切な安全装置、バックアップ、冗長性（redundancy）、その他の対策を講じることは使用者の責任となります。このソフトウェアもしくはハードウェアを危険が伴うアプリケーションで使用したことに起因して損害が発生しても、オラクル社およびその関連会社は一切の責任を負いかねます。

OracleおよびJavaはOracle Corporationおよびその関連企業の登録商標です。その他の名称は、それぞれの所有者の商標または登録商標です。

Intel, Intel Xeonは、Intel Corporationの商標または登録商標です。すべてのSPARCの商標はライセンスをもとに使用し、SPARC International, Inc.の商標または登録商標です。AMD, Opteron, AMDロゴ, AMD Opteronロゴは、Advanced Micro Devices, Inc.の商標または登録商標です。UNIXは、The Open Groupの登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。オラクル社およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

このドキュメントの使用	9
1 Oracle Solaris でのネットワーク管理について	11
Oracle Solaris ネットワークプロトコルスタックの説明	12
ハードウェアレイヤー	14
データリンクレイヤー	14
ネットワークレイヤー	15
トランスポートレイヤー	15
アプリケーションレイヤー	15
Oracle Solaris ネットワークプロトコルスタック内部のネームサービスとディレ クトリサービスの構成	16
Oracle Solaris のネットワークデバイスとデータリンク名	16
ネットワーク構成モードについて	17
固定モード	17
リアクティブモード	17
プロファイルベースのネットワーク構成について	18
Oracle Solaris ネットワーク管理コマンド	18
dladm コマンド	18
ipadm コマンド	19
route コマンド	20
netcfg および netadm コマンド	20
システムのネットワークを再構成するためのコマンド	20
ネットワーク上のクライアントシステムの構成に必要な情報	21
Oracle Solaris のネットワーク管理に関する詳細情報の参照先	22
2 Oracle Solaris でのデータリンク構成の管理	25
データリンク構成について	25
データリンクへの汎用名の割り当て	27
オペレーティングシステムによって汎用リンク名が割り当てられる方法のカスタ マイズ	28

アップグレードされたシステムでのリンク名	29
データリンクプロパティの管理	31
データリンクに関する一般情報の表示	32
システムのデータリンクの表示	32
データリンクの物理属性の表示	33
データリンクの削除	34
データリンクの名前の変更	35
データリンクのランタイム統計の取得	35
データリンクプロパティのカスタマイズ	36
ジャンボフレームのサポートの有効化	37
リンク速度パラメータの変更	37
データリンク上の STREAMS モジュールの設定	39
データリンクプロパティのステータス情報の取得	39
追加の dladm 構成タスク	41
▼ あるネットワークデバイスから別のデバイスへ IP 構成を移動する方法	41
▼ 動的再構成を使用してネットワークインタフェースカードを交換する方 法	43
▼ SPARC: 各インタフェースの MAC アドレスが一意であることを確認する 方法	45
3 Oracle Solaris での IP インタフェースとアドレスの構成および管理	49
ipadm コマンドを使用したネットワーク構成の管理	50
IPv4 インタフェースの構成	50
▼ IPv4 インタフェースを構成する方法	51
IPv6 インタフェースの構成	56
▼ IPv6 用にシステムを構成する方法	56
IPv6 インタフェースに対する一時アドレスの使用	59
IPv6 トークンの構成	61
サーバー上での IPv6 が有効なインタフェースの構成	64
IPv4 ネットワークから IPv6 ネットワークへの移行	65
ルーティングの構成	66
ルーティングテーブルとルーティングの種類	66
永続的 (静的) ルートの作成	68
単一インタフェースシステムのルーティングの有効化	71
IPv6 ルーティングについて	74
マルチホームホストの構成	75
▼ マルチホームホストの作成方法	76
マルチホームホストに対する対称ルーティングの実装	78
IP インタフェースプロパティとアドレスのカスタマイズ	79

MTU プロパティの設定	80
パケット転送の有効化	80
IP アドレスプロパティのカスタマイズ	81
IP インタフェース構成の無効化、削除、および変更	82
IP インタフェース構成の削除	83
IP インタフェース構成の無効化	83
IP インタフェース構成の削除または変更	84
IP インタフェースとアドレスのモニタリング	85
IP インタフェースに関する一般情報の取得	85
IP インタフェースに関する情報の取得	86
IP インタフェースのプロパティに関する情報の取得	87
IP アドレスに関する情報の取得	88
IP アドレスのプロパティに関する情報の取得	89
4 Oracle Solaris クライアントでのネームサービスとディレクトリサービスの管理	91
ネームサービス構成の新機能	91
ネームサービスとディレクトリサービスの構成の概要	92
name-service/switch SMF サービスについて	94
システムをローカルファイルモード用に構成する	95
▼ システムをローカルファイルモード用に構成する方法	96
DNS クライアントの構成	97
▼ DNS クライアントを有効にする方法	97
マルチキャスト DNS の有効化	99
DNS のためのリソースの通知	99
NIS クライアントの構成	100
▼ ブロードキャストモードで NIS クライアントを構成する方法	100
▼ 特定の NIS サーバーを使用して NIS クライアントを構成する方法	101
▼ NIS クライアントサービスを無効にする方法	102
LDAP クライアントの構成	102
ネームサービス構成のインポート	103
SMF ネームサービス構成のリセット	103
5 Oracle Solaris でのプロファイルベースのネットワーク構成の管理について	105
リアクティブモードについて	106
プロファイルベースのネットワーク構成について	107
プロファイルタイプの説明	107
システム定義およびユーザー定義のプロファイル	112
プロファイルベースのネットワーク構成の使用ガイドライン	113
プロファイルのアクティブ化ポリシー	113

プロファイルのアクティブ化条件	114
プロファイルベースのネットワーク構成を使用するためのセキュリティ要件	115
プロファイルベースのネットワーク構成がほかの Oracle Solaris 機能と連携して動作する方法	116
6 Oracle Solaris でのプロファイルベースのネットワーク構成の管理	119
プロファイルを有効および無効にする	119
プロファイルの構成	121
netcfg 対話型モードでの操作	122
netcfg コマンド行モードでの操作	123
netcfg コマンドファイルモードでの操作	124
NCP の作成	124
NCP に NCU を作成する	126
場所の作成	129
ENM の作成	131
既知の WLAN を作成する	133
プロファイルの管理	135
プロファイルのプロパティ値を設定する	135
プロファイル構成に関する情報を取得する	137
walkprop サブコマンドを使用してプロファイルのプロパティ値を設定する	141
プロファイルに関する情報を表示する	143
プロファイルを削除する	145
プロファイル構成をエクスポートする	146
エクスポートされたプロファイル構成を復元する	148
デスクトップからネットワーク構成を管理する	149
7 Oracle Solaris での無線ネットワークの管理	151
コマンド行を使用した無線ネットワークの管理	151
▼ WiFi ネットワークに接続する方法	151
▼ WiFi リンクをモニターする方法	156
セキュアな WiFi 通信の確立	157
▼ WEP キーを指定して暗号化された WiFi ネットワーク接続を設定する方法	158
リアクティブモードでの既知の WLAN の管理	159
デスクトップから無線ネットワークを管理する	160
▼ 無線ネットワークに加入する方法	160
デスクトップからお気に入りの無線ネットワークを管理する	161

索引 163

このドキュメントの使用

- **概要** – データリンク、IP インタフェースとアドレス、ネームサービスとディレクトリサービス、リアクティブプロファイル、無線ネットワークなど、Oracle Solaris オペレーティングシステム (OS) のさまざまなネットワークコンポーネントを構成および管理する方法に関する情報を提供します。
- **対象読者** – 企業のデータセンターでネットワーク構成の管理を担当するシステム管理者。
- **前提知識** – ネットワーク管理の概念と実践に関する基本的および高度な理解。

製品ドキュメントライブラリ

この製品の最新情報や既知の問題は、ドキュメントライブラリ (<http://www.oracle.com/pls/topic/lookup?ctx=E56342>) に含まれています。

Oracle サポートへのアクセス

Oracle のお客様は、My Oracle Support を通じて電子的なサポートを利用することができます。詳細は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> を参照してください。聴覚に障害をお持ちの場合は、<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> を参照してください。

フィードバック

このドキュメントに関するフィードバックを <http://www.oracle.com/goto/docfeedback> からお聞かせください。

◆◆◆ 第 1 章

Oracle Solaris でのネットワーク管理について

この章では、Oracle Solaris のホストクライアントシステムのネットワーク構成に含まれるさまざまなコンポーネントの概要について説明します。このガイドで説明するタスクおよび例では、インストール後にネットワーク構成を実行していることを想定しています。インストール時にネットワークを構成する手順については、『[Oracle Solaris 11.2 システムのインストール](#)』を参照してください。

ネットワークの新機能については、『[Oracle Solaris 11.2 新機能](#)』を参照してください。

ネットワーク上でクライアントシステムを構成する前に必要とされる計画タスクについては、『[Oracle Solaris 11.2 でのネットワーク配備の計画](#)』を参照してください。

よく使用されるネットワーク管理コマンドへのショートカットについては、『[Oracle Solaris 11.2 でのネットワーク管理の計画](#)』の第 3 章「ネットワーク管理コマンドのチートシート」を参照してください。

既存の TCP/IP ネットワークの管理については、『[Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理](#)』の第 1 章「TCP/IP ネットワークの管理」を参照してください。

この章の内容は、次のとおりです。

- 12 ページの「Oracle Solaris ネットワークプロトコルスタックの説明」
- 16 ページの「Oracle Solaris のネットワークデバイスとデータリンク名」
- 17 ページの「ネットワーク構成モードについて」
- 18 ページの「プロファイルベースのネットワーク構成について」
- 18 ページの「Oracle Solaris ネットワーク管理コマンド」
- 21 ページの「ネットワーク上のクライアントシステムの構成に必要な情報」
- 22 ページの「Oracle Solaris のネットワーク管理に関する詳細情報の参照先」

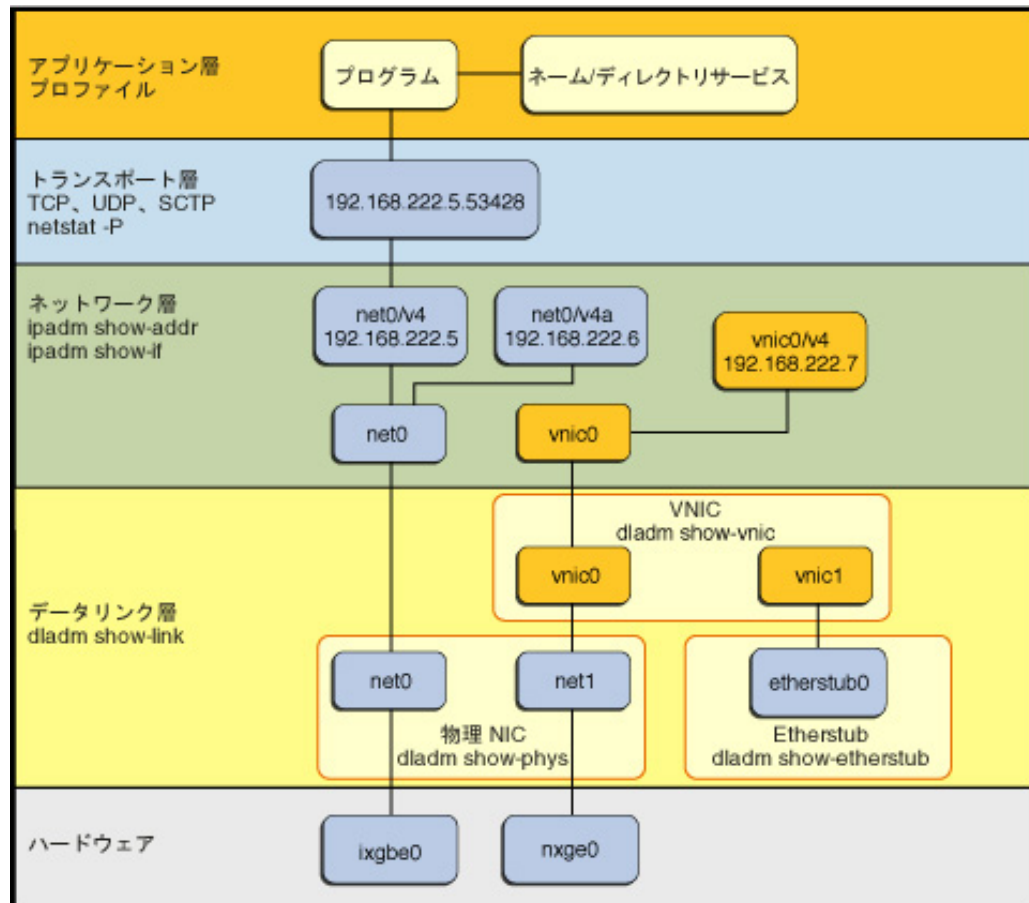
Oracle Solaris ネットワークプロトコルスタックの説明

ネットワークインタフェースは、システムとネットワーク間の接続を提供します。これらのインタフェースはデータリンク上に構成され、それがさらにシステム上のハードウェアデバイスのインスタンスに対応します。

Oracle Solaris 10 では、デバイス、データリンク、およびインタフェースをバインドする 1 対 1 の関係が存在しますが、これはネットワーク構成がハードウェア構成およびネットワークポロジに依存することを意味します。ネットワークインタフェースカード (NIC) の交換などの変更がハードウェアレイヤー内で実装された場合は、システム上でインタフェースを再構成する必要があります。

ただし、この Oracle Solaris リリースでは、物理データリンクの名前と、ネットワークデバイスに関連付けられたベースとなるハードウェアとの結び付きはなくなりました。デフォルトでは、図 1-1 に示すように、この種のデバイスには一般的な名前 `net`、およびシステム内のデバイスの物理位置を表す接尾辞が割り当てられます。この分離により、ネットワークレイヤーでのネットワーク構成は、ハードウェアレイヤーのチップセットやネットワークポロジにバインドされなくなりました。

図 1-1 Oracle Solaris 11 のネットワークプロトコルスタック



この実装によって、ネットワーク管理の柔軟性が次のように向上しています。

- ネットワーク構成が、ハードウェアレイヤーで発生する可能性のあるあらゆる変更から分離されます。ベースとなるハードウェアが取り外された場合でも、リンクおよびインタフェース構成は保持されます。そのため、2 枚の NIC のタイプが同じであれば、これらの同じ構成をどの交換用 NIC にも再適用できます。
- ネットワーク構成がネットワークハードウェア構成から分離されることにより、カスタマイズされたリンク名をデータリンクレイヤーで使用することもできます。
- データリンクレイヤーの抽象化によって、仮想ローカルエリアネットワーク (VLAN)、仮想ネットワークインタフェースカード (VNIC)、物理デバイス、リンクアグリゲーション、IP トンネルな

どの複数ネットワークの抽象化や構成が共通の管理エンティティ（つまり、データリンク）に統合されます。

Oracle Solaris 10 ネットワークスタックを Oracle Solaris 11 ネットワークスタックと比較する場合は、『[Oracle Solaris 10 から Oracle Solaris 11.2 への移行](#)』の「[Oracle Solaris 10 ネットワークプロトコルスタックと Oracle Solaris 11 ネットワークプロトコルスタックの比較](#)」を参照してください。

ハードウェアレイヤー

ネットワークハードウェアデバイスは、*ネットワークインタフェースカード (NIC)* または *ネットワークアダプタ* とも呼ばれます。NIC は、システムの購入時にシステムに組み込まれ、すでに存在している場合があります。または、別の NIC を購入してシステムに追加できます。特定の NIC は、カード上に存在する 1 つのインタフェースしか備えていません。その他の NIC ブランドは、さまざまなネットワーク操作を実行するように構成できる複数のインタフェースを備えていることがあります。

データリンクレイヤー

`dladm` コマンドを実行して、ネットワーク構成をデータリンクレイヤーで実行します。次に示すのは、このレイヤーで実行可能ないくつかのタイプのデータリンク構成です。

- 物理リンクの基本ネットワーク構成
- VNIC (物理リンク上の仮想リンク) の構成
 - VNIC 構成では、各仮想リンクは独自の MAC アドレスを保持します
- リンクアグリゲーション構成
 - アグリゲーションは、信頼性とパフォーマンスを目的として物理リンク上で構成されます。
- 仮想ローカルエリアネットワーク (VLAN) をサポートするための Etherstub
- VLAN をサポートするためのブリッジ

ほかの例については、『[Oracle Solaris 11.2 でのネットワークデータリンクの管理](#)』を参照してください。

ネットワークレイヤー

ipadm コマンドを実行して、ネットワーク構成をネットワークレイヤーで実行します。

次に示すのは、このレイヤーで実行可能ないくつかのタイプの IP 構成です。

- データリンク名と 1 対 1 で対応する名前を保持する IP インタフェース構成
- IP ネットワークマルチパス (IPMP)
- 仮想ネットワークインタフェース (VNI)
- 1 つの IP インタフェース用に複数の IP アドレス
- 単一の IP インタフェース上で構成された IPv4 および IPv6 アドレス
- アドレスオブジェクト名を使用して管理される IP アドレス

アドレスオブジェクト名は、インタフェース名とそれに続く一意の文字列で構成され、システムで構成された IP アドレスを表します。

ほかの例については、次を参照してください。

- 『Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理』
- 『Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理』
- 『Oracle Solaris ゾーンの作成と使用』

トランスポートレイヤー

トランスポートレイヤーで明示的な構成が必要になることはほとんどありません。通常、Oracle Solaris で動作するアプリケーションは、適切なトランスポートプロトコルおよび対応するポート番号を自動的に選択します。アクティブなポートは、netstat コマンドを使って表示できます。netstat(1M) のマニュアルページを参照してください。

いくつかのトランスポートプロトコルパラメータは、ipadm コマンドを使ってチューニングできます。詳細は、『Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理』の「トランスポート層サービスの管理」および『Oracle Solaris 11.2 カーネルのチューンアップ・リファレンスマニュアル』を参照してください。

アプリケーションレイヤー

アプリケーションプログラムは、ネットワークへのアクセスにソケット xti または tli アプリケーションプログラミングインタフェース (API) を使用します。これらの API では、接続の開始時

に、クライアントアプリケーションが対応するサーバーの IP アドレスおよびトランスポートポート番号を提供する必要があります。通常、サーバーは、IP アドレスではなく、リモートホストまたはサービス名でのみ知られています。アプリケーションは、標準ライブラリサービスを使用してホストおよびサービス名を IP アドレスに変換してから、ネットワーク接続を試みます。

詳細は、『[Oracle Solaris 11.2 Programming Interfaces Guide](#)』を参照してください。

Oracle Solaris ネットワークプロトコルスタック内部のネームサービスとディレクトリサービスの構成

クライアントシステムは、ネットワークの名前および IP アドレス検索を実行するように構成されているディレクトリサービスを保持している必要があります。ネットワーク環境によって、システムのブート時に構成プロセスが自動的に実行される場合と、手動での実行が必要な場合があります。この構成は、ネットワークスタックのアプリケーションレイヤーで行われます。

クライアント側のネームサービスおよびディレクトリサービスの構成オプションについては、[第4章「Oracle Solaris クライアントでのネームサービスとディレクトリサービスの管理」](#)を参照してください。

Oracle Solaris のネットワークデバイスとデータリンク名

データリンクは、開放型システム間相互接続 (OSI) モデルの 2 番目のレイヤー (L2) 内のリンクオブジェクトを表します。物理リンクはデバイスに直接関連付けられ、デバイス名を所有します。デバイス名には、ドライバ名およびデバイスインスタンス番号が含まれます。インスタンス番号には、システムでそのドライバを使用する NIC の数に応じて、0 - n の値を付けることができます。

デバイスインスタンス名は、システムのベースとなるハードウェアに引き続き依存します。ただし、ハードウェアレイヤーとソフトウェアレイヤーは分けられるため、これらのデバイスの上に構成されたデータリンクが同じようにバインドされることはなくなりました。このため、データリンクには、その構成のベースとなっているデバイス名以外の名前を割り当てることができます。

デフォルトでは、データリンクには net# 命名規則を使用する一般的な名前が割り当てられます (# はデバイスインスタンス番号)。インスタンス番号は、net0、net1、net2 などのように、システムのデバイスごとに増分します。詳しい概要については、[25 ページの「データリンク構成について」](#)を参照してください。

ネットワーク構成モードについて

Oracle Solaris では、固定とリアクティブの 2 つのネットワーク構成モードがサポートされています。

注記 - 固定モードおよびリアクティブモードという用語は、現在のネットワーク環境の変化に合わせて自動調整するシステムの機能を指し、これらのモードで IP アドレスを使用する際に静的または固定 IP アドレスを構成できるかどうかを指すものではありません。

固定モード

固定モードは、ネットワーク状態が変化するかどうかに関係なく、システム上のインスタンス化された構成が永続的であることを意味します。インタフェースの追加など、この種の変化が発生した場合は、新しいネットワーク環境に合わせてシステムのネットワークを再構成する必要があります。固定モードを使用する場合は、システムの構成に毎回同じネットワーク構成コマンドセットを使用します。企業のサーバーの場合、比較的ネットワーク環境が安定しているため、この構成モードがもっともよく使用されています。固定モードを使用する場合は、`dladm` および `ipadm` コマンドを使用して、ネットワーク構成のさまざまな側面を管理します。[18 ページの「Oracle Solaris ネットワーク管理コマンド」](#)を参照してください。

リアクティブモード

一方、リアクティブモードでは、現在のネットワーク状態に応じてネットワークが自動的に構成されます。このモードは、主にラップトップコンピュータやノートブックパーソナルコンピュータ (PC)、およびネットワーク状態が変化する可能性のある状況で使用されます。

リアクティブモードでは、ネットワークデーモン (`nwamd`) がシステムのネットワークインタフェースの状態をモニターします。ネットワーク状態が変化するときには、常にネットワークデーモンがネットワーク構成を動的に調整します。たとえば、ノートブック PC は、企業のネットワークに物理的に接続されている場合も、物理的に接続されていない場合もあります。物理的に接続されている場合は、たいいてい、ノートブックの無線インタフェースを無効にすることでしょう。また、ほとんどの場合に望ましいことは、Ethernet ケーブルがノートブックから取り外されたときに無線インタフェースを自動的に有効にすることです。さらに、無線ネットワークへの切り替え時に、システムが IP フィルタ設定を自動的に調整するようにもできます。リアクティブモードでは、ネットワークデーモンがこれらのタイプの動的構成変更を自動的に実行できます。一方、固定モードの場合には、これらのタイプの変更には手動による再構成が必要になります。

プロファイルベースのネットワーク構成について

プロファイルベースのネットワーク構成を使用すると、複数の代替構成を定義でき、それぞれが単一のプロファイル (ネットワーク構成プロファイル (NCP) と呼ばれる) で識別されます。たとえば、静的 IP アドレスおよび DNS サーバー場所を使用してシステムを構成する、ノートブック PC 用の office という名前のプロファイルを作成できます。代替の home プロファイルでは、DHCP を使用してこの情報を取得できます。単一のコマンドで、あるプロファイルから別のプロファイルに数秒間で切り替えることができます。有効化の可能なさまざまなタイプのプロファイルでは、固定とリアクティブの 2 つのネットワーク構成モードがサポートされます。デフォルトモードは、システムでどのプロファイルが現在アクティブになっているかで決まります。

Oracle Solaris のインストール時にシステム上でプロファイルをアクティブにする方法については、『[Oracle Solaris 10 から Oracle Solaris 11.2 への移行](#)』の「[インストール時のネットワークの構成動作](#)」を参照してください。

現在システムでアクティブなプロファイルが不明な場合は、`netadm list` コマンドを使用してこの情報を表示します。詳細は、[119 ページ](#)の「[プロファイルを有効および無効にする](#)」を参照してください。

Oracle Solaris でサポートされるさまざまなタイプのプロファイルの詳細は、[第5章「Oracle Solaris でのプロファイルベースのネットワーク構成の管理について」](#)を参照してください。

Oracle Solaris ネットワーク管理コマンド

次のコマンドを、ネットワーク構成を管理するために使用します。

- `dladm`
- `ipadm`
- `route`
- `netcfg`
- `netadm`

dladm コマンド

Oracle Solaris 10 で導入された `dladm` は、データリンクの構成に使用します。

dladm コマンドは、次のタイプのネットワーク構成の管理に使用します。

- 物理インタフェース – Ethernet、無線、および InfiniBand
- 仮想ネットワーク機能 – Etherstub、VNIC、および IP トンネル
- スイッチ機能 – リンクアグリゲーション、VLAN、およびブリッジテクノロジー
- デバイスの特性 – 速度、二重化、優先度、およびネゴシエーション機能

dladm コマンドは、システム上で現在アクティブなプロファイルの永続的ネットワーク構成を作成します。このため、net0 では、プロファイルが異なると MTU 値も異なることがあります。たとえば、net0 という名前のデータリンクが最大転送単位 (MTU) 1200 を使って構成されている場合、この MTU 値はそのプロファイルに限り net0 で永続的です。その後、別のプロファイルをアクティブにして、dladm コマンドを使用してそのプロファイルに異なる MTU 値を設定した場合、新しい MTU 値はそのプロファイルにのみ適用されます。[第2章「Oracle Solaris でのデータリンク構成の管理」](#)を参照してください。

dladm コマンドは、Oracle Solaris 10 でプロトコルプロパティの構成に使用される ndd コマンドも置き換えます。レイヤー 2 ドライバのプロパティを設定するためのツールとして、dladm コマンドには、ndd コマンドに優るいくつかの利点があります。[『Oracle Solaris 10 から Oracle Solaris 11.2 への移行』](#)の「[ndd コマンドと ipadm コマンドの比較](#)」を参照してください。

ipadm コマンド

このリリースでは、ipadm コマンドは、IP インタフェースおよびアドレス構成用の ifconfig コマンドを置き換えるものです。ipadm コマンドは、IP インタフェース管理専用であるため、IP インタフェースと IP アドレスをより効率的に管理します。また、ifconfig コマンドとは異なり、ipadm コマンドは永続的ネットワーク構成を実装しています。[『Oracle Solaris 10 から Oracle Solaris 11.2 への移行』](#)の「[ifconfig コマンドと ipadm コマンドの比較](#)」を参照してください。

ipadm コマンドは、Oracle Solaris 10 でプロトコルプロパティの構成に使用されていた ndd コマンドを置き換えるものでもあります。プロトコルのプロパティを設定するためのツールとして、ipadm コマンドには ndd に優るいくつかの利点があります。[『Oracle Solaris 10 から Oracle Solaris 11.2 への移行』](#)の「[ndd コマンドと ipadm コマンドの比較](#)」を参照してください。

route コマンド

/etc/defaultrouter ファイルは Oracle Solaris 11 では非推奨であるため、このファイルを使用してルート (デフォルトまたはそれ以外) を管理できなくなりました。代わりに、route コマンドを使用して手動でネットワークルーティングテーブルを操作してください。route コマンドは、アクティブなプロファイルのルートのみを操作します。デフォルトルート、およびほかのすべてのルートは、アクティブなプロファイルが変更された場合に置き換えられる可能性があります。システム上でプロファイルを切り替えない場合には、この問題は関係ありません。

詳細は、[68 ページの「永続的 \(静的\) ルートの作成」](#)を参照してください。

netcfg および netadm コマンド

netcfg および netadm コマンドは、さまざまなタイプのプロファイルの管理に使用されます。これら 2 つのコマンドで提供される機能の大半は、リアクティブプロファイルの管理が目的です。netcfg コマンドは、企業のサーバーではほとんど使用されません。通常、これらのタイプのサーバーは固定モードを使用します。

netadm コマンドは、プロファイルの有効化と無効化、およびプロファイルとその状態に関する情報の表示に使用されます。[119 ページの「プロファイルを有効および無効にする」](#)および[135 ページの「プロファイルの管理」](#)を参照してください。

注記 - 通常、リアクティブプロファイルのプロパティの構成には、netcfg コマンドが使用されません。ただし、プロファイルが現在アクティブである場合には、dladm および ipadm コマンドを使用して、リアクティブプロファイルの永続的構成を作成することもできます。ただし、netcfg コマンドを使用して、システム唯一の固定プロファイルである DefaultFixed を構成することはできません。詳細は、[netcfg\(1M\)](#) のマニュアルページを参照してください。

システムのネットワークを再構成するためのコマンド

システムのネットワークを再構成するために使用可能な別のオプションは、対話式システム構成 (SCI) ツールとも呼ばれる sysconfig ユーティリティです。SCI ツールは、新規インストールまたは構成解除されたシステムの構成をサポートするもので、テキストインストール時に新規作成される非大域ゾーン用のシステム構成を提供するように設計されています。SCI ツールは、対話的にも非対話的にも使用できます。

sysconfig ユーティリティを使用して、構成解除、構成、およびプロファイル作成という 3 つの操作を実行できます。unconfigure サブコマンドは、システム全体の構成解除に使用されます。このコマンドでは、システムは構成解除されたままになります。

configure サブコマンドは、システム全体またはシステムの一部を再構成するために使用され、これには次の 6 つの機能グループが含まれます。

- network
- location
- users
- identity
- support
- kdb_layout

構成解除されたグループのデフォルト値の詳細は、『[Oracle Solaris 11.2 システムのインストール](#)』の「[機能グループの概要](#)」を参照してください。

たとえば、次のようにシステムの既存のネームサービスを再構成します。

```
# sysconfig configure -g network,naming_services
```

-g オプションは、構成される特定の機能グループを指定するために使用します。この例では、システムのネットワークコンポーネントが構成されます。

詳細は、[sysconfig\(1M\)](#) のマニュアルページ、および『[Oracle Solaris 11.2 システムのインストール](#)』の第 6 章「[Oracle Solaris インスタンスの構成解除または再構成](#)」を参照してください。

ネットワーク上のクライアントシステムの構成に必要な情報

Oracle Solaris のインストール中、または Oracle Solaris のインストール後にネットワークを構成できます。このガイドでは、インストール後にネットワーク上でクライアントシステムを構成するためのタスクについて説明します。インストール時にネットワークを構成する手順については、『[Oracle Solaris 11.2 システムのインストール](#)』を参照してください。

ネットワーク上でクライアントシステムを構成するには、次の情報を指定する必要があります。

- ホスト名
- IP アドレス
- ネットマスク

詳細は、『Oracle Solaris 11.2 でのネットワーク配備の計画』の「ネットワークの IP アドレス指定形式の決定」および『Oracle Solaris 11.2 でのネットワーク配備の計画』の「ネットワークの IP 番号の取得」を参照してください。

ネットワークがサブネットに分割されている場合、各サブネットのシステムに適用するサブネット番号と IP アドレススキーマが必要です。それぞれのネットマスクも含まれます。『Oracle Solaris 11.2 でのネットワーク配備の計画』の「ネットワーク上のサブネットの使用」を参照してください。

- システムが属しているドメイン名

『Oracle Solaris 11.2 でのネットワーク配備の計画』の「ネットワーク上の名前付けエンティティの使用」を参照してください。

- デフォルトのルーターアドレス

この情報は、各ネットワークにルーターが 1 つしか接続していないような単純なネットワークポロジの場合、『Oracle Solaris 11.2 でのネットワーク配備の計画』の「ネットワーク上でルーターの計画」を参照してください。

またはルーターが RDISC (Router Discovery Protocol) や RIP (Routing Information Protocol) などのルーティングプロトコルを実行しない場合に指定します。Oracle Solaris でサポートされているルーターおよびルーティングプロトコルのリストの詳細は、『ルーターまたはロードバランサとしての Oracle Solaris 11.2 システムの構成』の「ルーティングプロトコル」を参照してください。

ネットワーク上でクライアントシステムを構成する場合は、『Oracle Solaris 11.2 でのネットワーク配備の計画』の「IPv4 自律システムのトポロジ」の情報を参照してください。

これらの各コンポーネントおよび関連するタスクの詳細は、『Oracle Solaris 11.2 でのネットワーク配備の計画』の第 1 章「ネットワーク配備の計画」を参照してください。

Oracle Solaris のネットワーク管理に関する詳細情報の参照先

このガイドに記載されている基本構成情報よりも詳しい情報については、ほかのタイプのネットワーク管理の実行に関する情報を次から参照してください。

- ネットワークプロトコルをカスタマイズしたり、トランスポートレイヤーサービスを管理したりする場合は、『Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理』の第 1 章「TCP/IP ネットワークの管理」を参照してください。
- システムをルーターまたはロードバランサとして構成する場合は、『ルーターまたはロードバランサとしての Oracle Solaris 11.2 システムの構成』を参照してください。

- リンクアグリゲーションやさまざまなタイプのブリッジなどの拡張データリンクおよび IP 構成を実行する場合は、『[Oracle Solaris 11.2 でのネットワークデータリンクの管理](#)』を参照してください。
- IPMP グループおよび IP トンネルを構成する場合は、『[Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理](#)』を参照してください。
- ネットワークのセキュリティーを確立するには、『[Oracle Solaris 11.2 でのネットワークのセキュリティー保護](#)』を参照してください。
- ネットワーク仮想化機能を実装する場合は、『[Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理](#)』を参照してください。

◆◆◆ 第 2 章

Oracle Solaris でのデータリンク構成の管理

この章では、ネットワークデバイスおよびデータリンクについて説明しますが、これには固定モード使用時の基本データリンク構成の管理タスクも含まれます。

次の情報は、物理リンクの構成またはネットワークデバイスを表すリンクにのみ関連します。構成可能なほかの Layer 2 (L2) エンティティーについては、『[Oracle Solaris 11.2 でのネットワークデータリンクの管理](#)』および『[Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理](#)』を参照してください。

この章の内容は、次のとおりです。

- 25 ページの「データリンク構成について」
- 31 ページの「データリンクプロパティの管理」
- 36 ページの「データリンクプロパティのカスタマイズ」
- 41 ページの「追加の `dladm` 構成タスク」

データリンク構成について

管理者が、データリンクの上に IP インタフェースを作成します。各データリンクは、開放型システム間相互接続 (OSI) モデルの 2 番目のレイヤー内のリンクオブジェクトを表します。データリンクは、物理ネットワークデバイス (物理リンクと呼ばれる)、物理データリンクのアグリゲーション、仮想ネットワークインタフェースカード (VNIC) などの多数の異なる L2 エンティティーを表すことができます。

リンク名は、関連付けられたリンクオブジェクトの自動作成時に割り当てられるか、データリンクの作成時に明示的にリンク名を割り当てることができます。物理リンク (物理ネットワークデバイスに関連付けられているリンク) は、デバイスの追加時、またはインストール後の Oracle Solaris システムの初回ブート時に自動的に作成されます。この Oracle Solaris リリースでは、物理データリンクの名前と、ネットワークデバイスに関連付けられたベースとなるハードウェアとの結び付きはなくなりました。デフォルトでは、データリンクに割り当てられる名前には、接

頭辞として `net` が、接尾辞としてシステム内のデータリンクの物理的な場所を示す番号が含まれます。たとえば、最初のオンボードネットワークデバイス `e1000g0` には名前 `net0` が割り当てられ、次の `e1000g1` デバイスには名前 `net1` が割り当てられる、という具合です。リンクアグリゲーションなど、明示的に作成したデータリンクに任意の名前を割り当てることができます。また、必要に応じて、デフォルトで割り当てられているデータリンクの `netN` 名を明示的に変更できます。

汎用または柔軟なリンク名には、ネットワーク構成に関して次の利点があります。

- 単一のシステムでは、動的再構成 (DR) が簡単になります。特定の NIC のためのネットワーク構成を、別の NIC 交換で継承できます。
- ゾーン移行のネットワーク設定に関する複雑さが軽減されました。移行先システムのリンクが、移行の前にそのゾーンに割り当てられていたリンクと同じ名前を共有している場合は、移行されたシステム内のゾーンでそのネットワーク構成が保持されます。そのため、移行のあと、そのゾーンでの追加のネットワーク構成は必要ありません。
- 汎用の命名規則により、インストール時に使用するシステム構成 (SC) マニフェストで指定されたネットワーク構成の複雑さが軽減されました。プライマリネットワークデータリンクには、一般的にすべてのシステムで名前 `net0` が付けられるため、`net0` 用の構成を指定する複数のシステムで一般的な SC マニフェストを使用できます。
- また、データリンクの管理も柔軟になります。データリンクで提供される特定の機能が反映されるように、データリンクの名前をさらにカスタマイズできます。

次の表は、ハードウェア (NIC)、デバイスインスタンス、リンク名、およびそのリンク上のインタフェースの間の新しい対応関係を示しています。データリンクの名前は、OS によって自動的に提供されます。

ハードウェア (NIC)	デバイスインスタンス	リンクの割り当てられた名前	IP インタフェース
<code>e1000g</code>	<code>e1000g0</code>	<code>net0</code>	<code>net0</code>
<code>igb</code>	<code>ixgbe</code>	<code>net1</code>	<code>net1</code>

この表に示すように、デバイスインスタンス名はハードウェアベースのままであるのに対し、データリンクの名前はインストール後に OS によって変更されます。

データリンク、その汎用名、および対応するデバイスインスタンス間のマッピングを表示するには、次のように `dladm show-phys` コマンドを使用します。

```
# dladm show-phys
LINK   MEDIA      STATE    SPEED    DUPLEX    DEVICE
net2   Ethernet   up       1000    full      bge2
```

```
net0    Ethernet    up    1000    full    e1000g0
net3    Ethernet    up    1000    full    nge3
net1    Ethernet    up    1000    full    e1000g1
```

データリンクへの汎用名の割り当て

- 物理ネットワークデバイスは、メディアタイプに従って順序が付けられ、特定のタイプが他よりも優先されます。メディアタイプは、次の降順の優先順位で順序が付けられています。

1. Ethernet
2. InfiniBand デバイス
3. Ethernet over IB
4. WiFi

- デバイスがメディアタイプに従ってグループ化およびソートされたあと、これらのデバイスは、その物理的な場所に基づいてさらに順序が付けられます。ここでは、システムボード上のデバイスが周辺デバイスより優先されます。

- そのメディアタイプと場所に基づいてより高い優先順位を持つデバイスに、より小さいインスタンス番号が割り当てられます。

SPARC ベースのシステムでは、OpenBoot PROM (OBP) で使用される netN デバイス別名に合わせて netN 名が割り当てられます。x86 ベースのシステムでは、SMBIOS データ (使用可能な場合) を使用してオンボードの Ethernet デバイスが識別され、それらに対して net0、net1、という具合に割り当てが行われます。これらの情報ソースに加え (または情報ソースが存在しない場合)、下位のマザーボードや IO ボード、ホストブリッジ、PCIe ルートコンプレックス、バス、デバイス、および機能上のデバイスは、ほかのデバイスに先立ってランク付けされ、上位のマザーボード、ホストブリッジなどのデバイスよりも下位の net インスタンスが割り当てられます。

リンク名、デバイス、および場所の対応を表示するには、次のように `dladm show-phys` コマンドと `-L` オプションを使用します。

```
# dladm show-phys -L
LINK          DEVICE          LOCATION
net0          e1000g0         MB
net1          e1000g1         MB
net2          e1000g2         MB
net3          e1000g3         MB
net4          ibp0            MB/RISER0/PCIE0/PORT1
net5          ibp1            MB/RISER0/PCIE0/PORT2
net6          eoib2           MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
net7          eoib4           MB/RISER0/PCIE0/PORT2/cloud-nm2gw-2/1A-ETH-2
```

オペレーティングシステムによって汎用リンク名が割り当てられる方法のカスタマイズ



注意 - 汎用リンク名が自動的に割り当てられる方法は、Oracle Solaris をインストールする前にカスタマイズする必要があります。インストール後は、既存の構成を削除しないかぎり、デフォルトのリンク名をカスタマイズすることはできません。

Oracle Solaris では、リンク名の割り当て時に接頭辞 `net` が使用されます。ただし、`eth` などの任意のカスタム接頭辞を使用できます。汎用リンク名の自動割り当てを無効にすることもできます。

リンク名の自動割り当てを無効にしたり、リンク名の接頭辞をカスタマイズしたりするには、システム構成 (SC) マニフェストで次のプロパティを設定します。SC マニフェストは、Oracle Solaris の自動インストーラ (AI) 機能によって使用されます。

```
<service name="network/datalink-management"
version="1" type="service">
<instance name="default enabled="true">
<property_group name='linkname-policy'
type='application'>
<propval name='phys-prefix' type='astring'
value='net' />
</property_group>
</instance>
</service
```

前の出力で太字で示されているように、`phys-prefix` プロパティの値は、デフォルトでは `net` に設定されます。

- 自動的な命名を無効にするには、次の例に示すように `phys-prefix` プロパティの値を空の文字列にします。

```
<propval name='phys-prefix' type='astring' value='' />
```

名前自動割り当てを無効にした場合、データリンク名は、関連付けられたハードウェアドライバ (`bge0` や `e1000g0` など) に基づきます。

- `net` 以外の接頭辞を使用するには、`phys-prefix` の値として (`eth` などの) 新しい接頭辞を指定します。

`phys-prefix` プロパティに指定された値が無効な場合、その値は無視されます。その場合、データリンクには、関連付けられたハードウェアドライバ (`bge0` や `e1000g0` など) に従って名前が付けられます。有効なリンク名に関する規則については、[30 ページの「有効なリンク名のための規則」](#)を参照してください。

アップグレードされたシステムでのリンク名

新規にインストールされたシステムでは、データリンクには `net0` から `netN-1` までの名前が自動的に付けられます。ここで、`N` はネットワークデバイスの総数を表します。

これに対して、別の Oracle Solaris 11 リリースからアップグレードした場合、データリンクでは、アップグレードの前に確立されていた名前が保持されます。これらの名前は、ハードウェアに基づくデフォルトの名前か、または管理者がアップグレードの前にデータリンクに割り当てたカスタマイズ名のどちらかです。さらに、これらのアップグレードされたシステムでは、そのあとに追加される新しいネットワークデバイスも、汎用名を受け取るのではなく、ハードウェアに基づくデフォルトの名前を保持します。アップグレードされたシステムでのこの動作により、OS によって割り当てられた汎用名が、その他のハードウェアに基づく名前や、アップグレードの前に管理者によって割り当てられたカスタマイズされた名前と混在しないことが保証されます。

両方のハードウェアベース名および OS で指定されたリンク名を、任意のほかの名前で置き換えることができます。通常、システムのネットワーク構成の作成には、OS が割り当てるデフォルトのリンク名で十分です。ただし、リンク名に変更を加える前に次の情報を考慮してください。

ハードウェアに基づくリンク名の置き換え

システムのリンクにハードウェアに基づく名前が付けられている場合は、これらのリンクの名前を少なくとも汎用名に変更します。ハードウェアに基づく名前を保持した場合は、あとでこれらの物理デバイスを取り外したり、交換したりしたときに、混乱が発生する可能性があります。

たとえば、デバイス `bge0` に関連付けられたリンク名 `bge0` を保持したとします。リンク構成はすべて、そのリンク名を参照することによって実行されます。あとで、NIC `bge` を NIC `e1000g` に交換したとします。以前のデバイスのリンク構成を新しい NIC `e1000g0` に再適用するには、リンク名 `bge0` を `e1000g0` に再割り当てする必要があります。ハードウェアに基づくリンク名 `bge0` を関連付けられた別の NIC `e1000g0` と組み合わせると、混乱が発生することがあります。ハードウェアに基づかない名前を使用することによって、そのリンクを関連付けられたデバイスからより適切に区別することができます。

リンク名の変更に関する注意事項

ハードウェアに基づくリンク名の置き換えはベストプラクティスですが、それでも、リンクの名前を変更する前には慎重な計画が必要です。デバイスのリンク名を変更しても、その新しい名前

が、既存の関連付けられたすべての構成に自動的に反映されるわけではありません。次の例は、リンク名を変更した場合のリスクを示しています。

- IP フィルタの構成にある一部の規則は、特定のリンクに適用されます。リンクの名前を変更すると、振り分け規則は引き続き、そのリンクの元の名前を参照します。その結果、リンクの名前を変更したあと、これらの規則は期待どおりに動作しなくなります。新しいリンク名を使用して、そのリンクに適用されるように振り分け規則を調整する必要があります。
- ネットワーク構成情報をエクスポートする可能性を考慮してください。前に説明したように、OS によって提供されるデフォルトの `net#` の名前を使用すると、ゾーンを移行し、ネットワーク構成を別のシステムに容易にエクスポートすることができます。ターゲットシステムのネットワークデバイスに `net0` や `net1` などの汎用名が付けられている場合、これらのゾーンは単純に、そのゾーンに割り当てられたデータリンクに名前が一致するデータリンクのネットワーク構成を継承します。

そのため、一般的な規則として、データリンクの名前はランダムに変更しないでください。データリンクの名前を変更する場合は、そのリンクの関連付けられたすべての構成が、リンク名が変更されたあとも引き続き適用されることを確認してください。

リンクの名前を変更することによって影響を受ける可能性のある一部の構成は、次のとおりです。

- IP 振り分け規則
- `ipadm` コマンドを使用して指定された IP 構成
- Oracle Solaris 11 ゾーン
- `autopush` 構成

注記 - リンクの名前を変更したときに `autopush` 構成を変更する必要はありません。ただし、リンクの名前が変更されたあと、この構成がリンクごとの `autopush` プロパティでどのように動作するかを認識しておく必要があります。詳細は、[データリンク上の STREAMS モジュールの設定](#)を参照してください。

有効なリンク名のための規則

リンク名を割り当てる場合は、次の規則に従ってください。

- リンク名は、文字列と物理接続点 (PPA) 番号で構成する必要があります。
- リンク名は、次の制約に従う必要があります。

- 名前は 3 文字から 8 文字までの間で構成されることが理想です。ただし、名前の最大文字数は 16 です。
- 名前のための有効な文字は、英数字 (a-z, 0-9) とアンダースコア (_) です。



注意 - リンク名には大文字を使用しないでください。

- 各データリンクに割り当てるリンク名は、一度に 1 つだけにする必要があります。
- 各データリンクには、システム内で一意のリンク名を割り当てる必要があります。

注記 - 追加の制限として、`lo0` は柔軟なリンク名としては使用できません。この名前は、IP ループバックインタフェースを識別するために予約されています。

リンク名を割り当てるときに、ネットワーク設定内のリンクの機能が役立つリファレンスになる場合があります。たとえば、`netmgt0` をネットワーク管理専用のリンクにすることができます。Upstream2 を ISP に接続するリンクにすることができます。混乱を回避するための一般的な規則として、既知のデバイスの名前をリンクに割り当てることはしないでください。

データリンクプロパティの管理

`dladm` コマンドを使用して一般的なデータリンクプロパティをカスタマイズすると、次の利点があります。

- `dladm` コマンドは、ネットワークドライバプロパティの構成に必要な唯一のコマンドインタフェースです。このコマンドは、`ndd` コマンドと `driver.conf` ファイルの変更を組み合わせ使用して、ドライバプロパティを設定するという以前の用法に代わるものです。
- 設定するプロパティにかかわらず、次の一貫した構文が使用されます。

```
dladm subcommand properties datalink
```

- `dladm` コマンドの使用は、ドライバの公開プロパティと非公開プロパティの両方に適用されます。
- 特定のドライバに対して `dladm` コマンドを使用しても、似たタイプのほかの NIC のネットワーク接続に支障が生じることはありません。したがって、データリンクのプロパティを動的に構成できます。
- データリンク構成の値は `dladm` リポジトリ内に格納され、システムのリブート後も永続します。

データリンクに関する一般情報の表示

`dladm` コマンドをオプションなしで使用すると、システムのデータリンクに関する一般情報 (クラス、状態、およびベースとなる物理リンクを含む) が表示されます。

```
# dladm
LINK      CLASS    MTU     STATE   OVER
net0      phys     1500    unknown --
net1      phys     1500    up      --
net2      phys     1500    unknown --
net3      phys     1500    unknown --
net4      phys     1500    up      --
aggr0     aggr     1500    up      net1,net4
```

データリンクは、物理リンク以外のさまざまなクラス (たとえば、リンクアグリゲーション、仮想 LAN (VLAN)、仮想 NIC (VNIC)) である可能性があります。これらのほかのデータリンクは、`dladm` コマンドで表示されるデフォルトの情報にも含まれます。たとえば、前の出力では、リンクアグリゲーション `aggr0` は物理データリンク `net1` と `net4` 上に構成されています。

リンクアグリゲーションと VLAN については、『[Oracle Solaris 11.2 でのネットワークデータリンクの管理](#)』を参照してください。VNIC については、『[Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理](#)』を参照してください。

システムのデータリンクの表示

`dladm show-link` コマンドを使用して、システム上の物理データリンクと仮想データリンクの両方を表示します。システムには、取り付けられた NIC と同じ数のデータリンクがあります。このコマンドでさまざまなオプションを使用して、表示される情報をカスタマイズできます。

追加のオプションや引数なしで使用する場合、`dladm show-link` コマンドは次の情報を表示します。

```
# dladm show-link
LINK      CLASS    MTU     STATE   OVER
net1      phys     1500    down    --
net3      phys     1500    unknown --
net0      phys     1500    up      --
net2      phys     1500    unknown --
net11     phys     1500    up      --
net5      phys     1500    up      --
net6      phys     1500    up      --
```


前の出力では、STATE 列は仮想データリンクの現在の状態を示します。この状態は、up、down、unknown のいずれかになります。仮想データリンクの場合、NIC が複数の VNIC に分割されると、内部で仮想スイッチが暗黙的に作成されます。この仮想スイッチの作成により、物理データリンクが外部ネットワークに接続されていなくても、VNIC およびプライマリデータリンクが同一の VLAN 上に存在するかぎり、相互の通信が可能になります。この関係により、データリンクの仮想状態が形成されます。

-P オプションを使用して、データリンクに関する永続的な構成情報を表示します。このコマンドで表示された情報に基づいて、その後のネットワーク構成を進めることができます。たとえば、システム上の NIC の数を決定したり、IP インタフェースを構成するときに使用するデータリンクを選択したりできます。このコマンドを入力すると、表示される情報は次の例のようになります。

```
# dladm show-link -P
LINK      CLASS    OVER
net0      phys     --
net1      phys     --
net2      phys     --
```

前の例では、システムに、対応する物理 NIC に直接関連付けられた 3 つのデータリンクが表示されています。phys クラスに属するデータリンク上に構成された特殊なデータリンク (集積体や仮想 NIC など) は存在しません。

データリンクの物理属性の表示

システムのデータリンクに関連付けられている物理 NIC に関する情報を表示するには、dladm show-phys コマンドを使用します。このコマンドをオプションなしで使用すると、次の例のような情報が表示されます。

```
# dladm show-phys
LINK      MEDIA      STATE    SPEED    DUPLEX    DEVICE
net0      Ethernet   up       100Mb    full     e1000g0
net1      Ethernet   down     0Mb      --       nge0
net2      Ethernet   up       100Mb    full     bge0
net3      InfiniBand --        0Mb      --       ibd0
```

前の出力には、ほかの詳細とともに、汎用リンク名を持つデータリンクが関連付けられている物理 NIC が表示されています。たとえば、net0 は NIC e1000g0 のデータリンク名です。データリンクに設定されたフラグに関する情報を表示するには、-P オプションを使用します。たとえば、r というフラグが設定されたデータリンクは、そのベースとなる NIC が取り外されたことを意味します。

前の出力では、STATE 列は物理データリンクの現在の状態を示します。この状態は、up、down、unknown のいずれかになります。物理リンク状態は、物理デバイスに外部ネットワークとの接続があるかどうか (ケーブルが接続され、そのケーブルの反対側にあるポートの状態が up の場合は接続がある) を識別します。

-L オプションは、使用可能な別の有用なオプションです。このオプションは、各データリンクの物理的な場所を表示します。この場所によって、データリンクのインスタンス番号 (net0、net1 など) が決まります。

```
# dladm show-phys -L
LINK    DEVICE    LOCATION
net0    bge0      MB
net2    ibp0      MB/RISER0/PCIE0/PORT1
net3    ibp1      MB/RISER0/PCIE0/PORT2
net4    eoib2     MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

-m オプションを使用して、システム内の物理リンクの MAC アドレスを表示します。

```
# dladm show-phys -m
LINK          SLOT    ADDRESS          INUSE CLIENT
net0          primary 0:11:22:a9:ee:66  yes  net0
```

このコマンドは、ifconfig コマンドと同様に使用します。

システム内のすべてのリンク (物理および物理以外) の MAC アドレスは次のように表示します。

```
# dladm show-linkprop -p mac-address
LINK    PROPERTY    PERM VALUE          EFFECTIVE    DEFAULT    POSSIBLE
net0    mac-address  rw  0:11:22:a9:ee:66  0:11:22:a9:ee:66  0:11:22:a9:ee:66
--
```

データリンクの削除

システムからデータリンクを削除するには、`dladm delete-phys` コマンドを使用します。

データリンクの削除は、物理 NIC の削除と緩やかに結び付いています。たとえば、システムから物理 NIC が取り外される場合、『[Oracle Solaris 10 から Oracle Solaris 11.2 への移行](#)』の「[Oracle Solaris 10 ネットワークプロトコルスタックと Oracle Solaris 11 ネットワークプロトコルスタックの比較](#)」で説明されているように、ソフトウェアレイヤーはもはやハードウェアレイヤーにバインドされていないため、その NIC に関連付けられたデータリンク構成は残ります。したがって、そのデータリンクの名前をほかの NIC に関連付けられたリンクに割り当てることにより、別のベースとなる物理 NIC でこのデータリンク構成を引き続き使用できます。

NIC を交換せずに切り離し、そのデータリンク構成が不要になった場合は、次のようにしてデータリンクを削除できます。

```
# dladm delete-phys datalink
```

ヒント - データリンクの NIC が削除されたかどうかを確認するには、`dladm show-phys -P` コマンドを使用します。出力に含まれる `FLAGS` 列の `r` フラグは、物理リンクに関連付けられた物理デバイスが取り外されたかどうかを示します。

データリンクの名前の変更

データリンクの名前を変更するには、`dladm rename-link` コマンドを使用します。Oracle Solaris システムでは、OS がすべてのデータリンクに対して自動的に汎用名を設定します。汎用データリンク名の詳細は、[25 ページの「データリンク構成について」](#)を参照してください。

これらの汎用名には、デフォルトで `netn` (`net0`, `net1`, `net2` など) という命名形式が使用されます。OS がこれらの名前を管理しているため、通常の実務の一部としてデータリンクの名前を変更することはありません。リンク名の変更が必要な手順については、[41 ページの「あるネットワークデバイスから別のデバイスへ IP 構成を移動する方法」](#)を参照してください。

データリンクのランタイム統計の取得

`dlstat` コマンドを使用して、すべてのデータリンクタイプのランタイムデータリンク統計を取得します。`dlstat` をオプションなしで単独で使用する場合は、次の出力に示すように、システム上のすべてのデータリンクに関する統計情報が表示されます。

```
% dladm dlstat
LINK      IPKTS      RBYTES      OPKTS      OBYTES
net0      58.00K      9.52M      5.61K      1.91M
```

`dlstat` コマンドの使用に関する詳細は、『[Oracle Solaris 11.2 での仮想ネットワークとネットワークリソースの管理](#)』の第 8 章「[ネットワークトラフィックとリソース使用状況のモニタリング](#)」を参照してください。[dlstat\(1M\)](#) のマニュアルページも参照してください。

データリンクプロパティのカスタマイズ

基本的なデータリンク構成に加えて、`dladm` コマンドを使用してデータリンクプロパティを設定し、ネットワークの要件に応じてそれらをカスタマイズすることもできます。

次の 3 つの `dladm` サブコマンドを使用して、データリンクプロパティを管理します。

`dladm show-linkprop -p property data-link` データリンクのプロパティおよびその現在値を表示します。`-p property` オプションを使用しなかった場合は、データリンクのすべてのプロパティが表示されます。データリンクを指定しなかった場合は、すべてのデータリンクのすべてのプロパティが表示されます。

`dladm set-linkprop -p property=value data-link` 値をデータリンクのプロパティに割り当てます。

`dladm reset-linkprop -p property data-link` データリンクの特定のプロパティをデフォルト値にリセットします。

カスタマイズ可能なデータリンクプロパティは、ある特定の NIC ドライバがサポートするプロパティに依存します。

`dladm` コマンドを使用して構成可能なデータリンクプロパティは、2 つのカテゴリのいずれかに該当します。

- 公開プロパティ – これらのプロパティは、Ethernet 用のリンク速度や自動ネゴシエーション、またはすべてのデータリンクドライバに適用可能な最大転送単位 (MTU) サイズなど、特定メディアタイプの任意のドライバに適用できます。
- 非公開プロパティ – これらのプロパティは、特定メディアタイプの NIC ドライバの特定のサブセットに固有です。これらのプロパティはそのサブセットに固有のものとなる可能性があります。それは、これらのプロパティが、ドライバに関連付けられたハードウェアに密接に関連しているか、あるいはデバッグ関連のチューニング可能パラメータのように、ドライバ実装自体の詳細に密接に関連しているからです。

リンクプロパティには通常、デフォルト値があります。ただし、特定のネットワークシナリオでは、特定のプロパティの値を変更する必要がある場合もあります。たとえば、NIC は、自動ネゴシエーションを正しく実行しない古いスイッチと通信する可能性があります。また、スイッチは、ジャンボフレームをサポートするように構成されている可能性があります。または、特定のドライバで、パケットの送信やパケットの受信を制御するドライバ固有のプロパティを変更する必要がある可能性もあります。

ジャンボフレームのサポートの有効化

MTU は、あるプロトコルによってシステムから転送できるパケットの最大サイズを定義します。大部分の NIC ドライバでは、MTU サイズがデフォルトで 1500 に定義されています。しかし、ジャンボフレームがネットワークを通過する場合は、デフォルト値では不十分です。ジャンボフレームをサポートするには、MTU サイズを少なくとも 9000 にする必要があります。

注記 - MTU プロパティはデータリンクと IP インタフェースの両方に共通であり、これはデータリンク用のある MTU 値と、そのリンク経由で構成された IP インタフェース用の別の MTU 値を保持できることを意味します。データリンク MTU の値は、IP インタフェースの MTU に設定可能な値に影響を及ぼします。データリンクと IP インタフェースの MTU プロパティを構成する場合に、この動作が意味することについて詳しくは、[80 ページの「MTU プロパティの設定」](#)を参照してください。

次のように、MTU サイズのデフォルト値を変更します。

```
# dladm set-linkprop -p mtu=new-size datalink
```

MTU サイズを変更したあとは、データリンク上の IP インタフェースを再構成できます。

次の例は、ジャンボフレームのサポートを有効にする方法を示しています。この例は、データリンク上の既存の IP インタフェース構成がすでに削除されていることを前提にしています。

```
# dladm show-linkprop -p mtu net1
LINK    PROPERTY    PERM VALUE    EFFECTIVE    DEFAULT    POSSIBLE
net1    mtu         rw  1500        1500        1500        1500

# dladm set-linkprop -p mtu=9000 net1
# dladm show-link net1
LINK    CLASS    MTU    STATE    BRIDGE    OVER
web1    phys    9000    up      --      --
```

リンク速度パラメータの変更

ほとんどのネットワーク設定は、異なる速度機能を備えたシステムの組み合わせから構成されます。各システムは、ネットワーク内のほかのシステムに速度機能を通知して、各システムがネットワークトラフィックをどのように送受信するかを知らせます。

次の対になったデータリンクプロパティは、システムによって通知される速度機能を制御します。

```
■ adv_10gfdx_cap/en_10gfdx_cap
```

- `adv_1000fdx_cap/en_1000fdx_cap`
- `adv_1000hdx_cap/en_1000hdx_cap`
- `adv_100fdx_cap/en_100fdx_cap`
- `adv_100hdx_cap/en_100hdx_cap`
- `adv_10fdx_cap/en_10fdx_cap`
- `adv_10hdx_cap/en_10hdx_cap`

それぞれのリンク速度機能は、通知速度 (`adv*_cap`) と有効化通知速度 (`en*_cap`) から成る一組のプロパティで参照されます。さらに、プロパティ名に含まれる `*fdx*` と `*hdx*` が示すように、全二重機能と半二重機能のどちらに対してもデータリンク速度の情報が提供されます。通知速度プロパティは、特定のデータリンク速度が通知されるかどうかを示す読み取り専用プロパティです。特定のデータリンク速度が通知されるかどうかを決定するには、対応する `en*_cap` プロパティを設定します。

デフォルトでは、データリンクのすべての速度機能と二重機能が通知されます。しかし、新しいシステムが古いシステムと通信していて、自動ネゴシエーションが無効または未サポートになっている場合もあります。これらの 2 つのシステム間の通信を有効にするには、古いシステムと新しいシステム間の通知速度を低い値に変更する必要がある場合があります。場合によっては、システムのギガビット機能をオフにする必要があり、低速な速度機能のみが通知されます。この例では、次のコマンドを入力して、全二重機能と半二重機能の両方でギガビット機能の通知をオフに切り替えます。

```
# dladm set-linkprop -p en_1000fdx_cap=0 datalink
# dladm set-linkprop -p en_1000hdx_cap=0 datalink
```

これらのプロパティの新しい値を表示するには、次のように `dladm show-linkprop` コマンドを使用します。

```
# dladm show-linkprop -p adv_10gfdx_cap datalink
# dladm show-linkprop -p adv_1000hdx_cap datalink
```

通常、特定の有効化速度プロパティと対応する通知プロパティの値は同一です。ただし、NIC が Power Management などの一部の高度な機能をサポートしている場合は、それらの機能が、ホストとそのリンクパートナーとの間で実際に通知されるビット数の制限を設定する可能性があります。たとえば、Power Management を使用する場合、`adv*_cap` プロパティの設定が `en*_cap` プロパティの設定のサブセットにすぎなくなる可能性があります。

データリンク上の STREAMS モジュールの設定

データリンクが開かれたときにストリーム上にプッシュされる STREAMS モジュールを、最大 8 個まで設定できます。これらのモジュールは通常、仮想プライベートネットワーク (VPN) やファイアウォールなど、他社製のネットワークソフトウェアによって使用されます。そのようなネットワークソフトウェアに関するドキュメントは、ソフトウェアベンダーから提供されています。

特定のデータリンク上にプッシュするモジュールのリストは、`autopush` プロパティによって制御されます。また、`autopush` プロパティの値は、`dladm set-linkprop` コマンドを使用することによって設定されます。

データリンクのストリーム上にモジュールをドライバ単位でプッシュするために使用可能な、別個の `autopush` コマンドが存在します。このコマンドは、ドライバごとに設定された構成ファイルを使用して、プッシュするモジュールの情報を取得します。ただし、ドライバは常に NIC にバインドされます。データリンクのベースとなる NIC が取り外されると、そのリンクの `autopush` プロパティの情報も失われます。

したがって、このためには `autopush` コマンドよりも `dladm` コマンドを使用することをお勧めします。特定のデータリンクでドライバ単位とリンク単位の両方のタイプの `autopush` 構成が存在している場合は、`dladm set-linkprop` コマンドで設定されたリンク単位の情報が使用され、ドライバ単位の情報は無視されます。

データリンクが開かれたときに STREAMS にモジュールをプッシュするには、同じ `dladm set-linkprop` コマンドを使用して、`autopush` プロパティのモジュールを指定します。たとえば、次のように `vpnmod` および `bufmod` モジュールをリンク `net0` の上にプッシュします。

```
# dladm set-linkprop -p autopush=vpnmod.bufmod net0
```

データリンクプロパティのステータス情報の取得

データリンクプロパティに関する情報を取得するには、次のいずれかのコマンドを使用できます。

- `dladm show-linkprop -p property datalink`
- `dladm show-ether datalink`

データリンクプロパティの表示

データリンクプロパティの完全なリストを表示するには、次の例に示すように、コマンドをプロパティを指定せずに入力します。

```
# dladm show-linkprop net1
LINK      PROPERTY      PERM VALUE      EFFECTIVE      DEFAULT      POSSIBLE
net1      speed          r-  0            0              0            --
net1      autopush       rw  --           --            --           --
net1      zone           rw  --           --            --           --
net1      duplex         r-  unknown       unknown        unknown      half,full
net1      state          r-  up             up             up           up,down
net1      adv_autoneg_cap --  --           --            0            1,0
net1      mtu            rw  1500          1500          1500        1500
net1      flowctrl       --  --           --            no           no,tx,rx,bi,
                                pfc,auto
net1      adv_10gfdx_cap r-  --           --            0            1,0
net1      en_10gfdx_cap --  --           --            0            1,0
net1      adv_1000fdx_cap r-  --           --            0            1,0
net1      en_1000fdx_cap --  --           --            0            1,0
net1      adv_1000hdx_cap r-  --           --            0            1,0
net1      en_1000hdx_cap --  --           --            0            1,0
net1      adv_100fdx_cap r-  --           --            0            1,0
net1      en_100fdx_cap --  --           --            0            1,0
net1      adv_100hdx_cap r-  --           --            0            1,0
net1      en_100hdx_cap --  --           --            0            1,0
net1      adv_10fdx_cap r-  --           --            0            1,0
net1      en_10fdx_cap  --  --           --            0            1,0
net1      adv_10hdx_cap r-  --           --            0            1,0
net1      en_10hdx_cap  --  --           --            0            1,0
net1      maxbw          rw  --           --            --           --
net1      cpus           rw  --           --            --           --
```

Ethernet プロパティ値の表示

`dladm show-ether` コマンドにオプションを何も指定しない場合、データリンクの現在の Ethernet プロパティ値だけが表示されます。デフォルトで提供されるもの以外の情報を取得するには、次の例に示すように `-x` オプションを使用します。

```
# dladm show-ether -x net1
LINK      PTYPE      STATE      AUTO      SPEED-DUPLEX      PAUSE
net1      current    up         yes       1G-f              both
--      capable    --         yes       1G-fh,100M-fh,10M-fh both
--      adv        --         yes       100M-fh,10M-fh   both
--      peeradv    --         yes       100M-f,10M-f     both
```

`-x` オプションを使用すると、このコマンドは指定されたリンクの組み込み機能や、ホストとリンクパートナーとの間で現在通知されている機能も表示します。

この例では、次の情報が表示されます。

- Ethernet デバイスの現在の状態については、リンクは稼働しており、毎秒 1 ギガビットの全二重で機能しています。その自動ネゴシエーション機能が有効化されており、ホストとリンクパートナーの両方が一時停止フレームを送受信できる双方向フロー制御が備わっています。この情報は、出力の 1 行目に表示されています。
- サンプル出力の後続の行には、データリンク速度機能に関する情報、通知される実際のデータリンク速度、およびピアシステムからの情報が次のように表示されています。
 - Ethernet デバイスの機能が一覧表示されています。ネゴシエーションタイプは自動に設定できます。さらに、このデバイスは毎秒 1 ギガビット、毎秒 100 メガビット、および毎秒 10 メガビットの速度を、全二重、半二重の両方でサポートできます。同様に、一時停止フレームはホストとリンクパートナーとの間でどちらの方向にも送受信できます。
 - net1 の機能が次のように通知されています: 自動ネゴシエーション、速度 - 二重、一時停止フレームのフロー制御。
 - 同様に、net1 のリンクパートナーつまりピアパートナーが、次の機能を通知しています: 自動ネゴシエーション、速度 - 二重、一時停止フレームのフロー制御。

追加の dladm 構成タスク

このセクションでは、プライマリインタフェースの切り替えや動的再構成 (DR) の実行など、dladm コマンドを使用することで単純化されたその他の構成手順について説明します。

▼ あるネットワークデバイスから別のデバイスへ IP 構成を移動する方法

あるネットワークデバイスに関連付けられている IP 構成を保存してから、その構成を別のネットワークデバイスに移動する必要がある場合は、次の手順を使用します。この手順は、システムからネットワークカードを取り外すための準備として、またはネットワークケーブル接続を変更する際に実行できます。

あくまでも例として、この手順では net0 (e1000g0) デバイスに関連付けられている IP 構成を保存し、その構成を nge0 デバイスに適用する方法を説明します。

1. 管理者になります。

2. システム上のデバイスマッピングへの物理リンクを表示します。

次の例では、e1000g0 の IP 構成が何らかの理由で停止し、そのために構成を nge0 に移動する必要があるものとします。

```
# dladm show-phys
LINK   MEDIA   STATE   SPEED   DUPLEX   DEVICE
net0   Ethernet down    0       unknown e1000g0
net1   Ethernet down    0       unknown e1000g1
net2   Ethernet up     1000    full    nge0
net3   Ethernet down    0       unknown nge1
```

3. データリンク上の IP 構成を一時的に無効にし、その永続的な設定はそのままにします。

```
# ipadm disable-if interface
```

たとえば、次のように net0 上の IP 構成を無効にします。

```
# ipadm disable-if net0
```

この手順を使用すると、IP 構成を再作成せずにデータリンクの名前を変更できます。

4. データリンクの名前を変更します。

たとえば、次のように net0 データリンクの名前を変更できます。

```
# dladm rename-link net0 oldnet0
```

5. プライマリリンク名を、プライマリデバイスになるように指定されたデータリンクに割り当てます。

```
# dladm rename-link new-link primary-link
```

たとえば、次のようにして、net0 リンク名を net2 データリンクに再割り当てします。

```
# dladm rename-link net2 net0
```

6. 新しい名前のデータリンク上で IP 構成を再度有効にします。例:

```
# ipadm enable-if -t net0
```

例 2-1 データリンクインタフェースの削除

新規インストールを実行する場合は、システム上のネットワークデバイスの総数に応じて net0、net1、netN と続く命名規則を使用する総称名が、システムのすべてのデータリンクに自動的に割り当てられます。インストール後に、データリンクに異なる名前を割り当てることができます。次の例は、インタフェースに対して最初に提供された IP アドレスを変更する方法を示します (最初に既存のインタフェースを削除する操作を含む)。

```
# ipadm delete-ip net0
# ipadm create-ip net0
# ipadm create-addr -T addrconf net0/new-add
```

詳細は、[Chapter 2, Oracle Solaris](#) でのデータリンク構成の管理を参照してください。

▼ 動的再構成を使用してネットワークインタフェースカードを交換する方法

次の手順は、動的再構成 (DR) をサポートするシステムにのみ適用されます。特に、DR の完了後の構成手順を示します。DR プロセスの完了後にネットワークリンクを構成し直す必要はなくなりました。代わりに、取り外した NIC のリンク構成を交換した NIC に移動するだけです。

この手順では、DR 自体の実行手順は示しません。この情報については、使用しているシステムドキュメントを参照してください。

DR の紹介については、『[Oracle Solaris 11.2 でのデバイスの管理](#)』の第 2 章「[デバイスの動的構成](#)」を参照してください。

始める前に 最初に次の手順を完了していることを確認します。

- システムが DR をサポートしていることを確認します。
- システムでの DR を説明した適切なマニュアルを参照します。

Oracle 製 Sun サーバーの DR に関する最新ドキュメントを見つけるには、<http://www.oracle.com/technetwork/indexes/documentation/index.html> で「動的再構成 (dynamic reconfiguration)」を検索します。

Oracle Solaris Cluster 環境での DR の実行については、『[Oracle Solaris Cluster システム管理](#)』を参照してください。

1. 管理者になります。
2. (オプション) データリンクの物理属性や各データリンクのシステム上の場所に関する情報を表示します。

```
# dladm show-phys -L
```

`dladm show-phys -L` コマンドで表示される情報タイプの詳細は、[dladm\(1M\)](#) のマニュアルページを参照してください。

3. システムのドキュメントの説明に従って、DR プロセスを実行します。

4. 交換用の NIC を取り付けたあとで、該当する状況に応じて、次のように手順を実行します。

■ 古い NIC と同じスロットに交換用 NIC を挿入した場合は、手順 5 に進みます。

古い NIC が以前に占有していた場所を新しい NIC が使用しているため、新しい NIC が古い NIC のリンク名と構成を継承します。

■ 交換用 NIC を別のスロットに挿入し、新しい NIC が取り外された NIC のデータリンク構成を継承する必要がある場合は、次のようにリンクの名前を変更します。

```
# dladm rename-link new-datalink old-datalink
```

new-datalink 古い NIC が取り外された場所とは異なるスロットに挿入された交換用 NIC のデータリンクを示します。

old-datalink 取り外された古い NIC に関連付けられているデータリンク名を示します。

注記 - このシナリオでは、古い NIC が取り外されたスロットは空けておく必要があります。

たとえば、スロット 1 の NIC が取り外され、スロット 2 に新しい NIC が挿入されます。スロット 1 には NIC が挿入されていません。スロット 1 上のデータリンクが `net0` で、スロット 2 上のデータリンクが `net1` であるとします。次のようにして、新しい NIC のデータリンクが古い NIC のデータリンク構成を継承することを指定します。

```
# dladm rename-link net1 net0
```

5. 新しい NIC リソースが使用可能になるようにして、DR プロセスを完了します。

たとえば、`cfgadm` コマンドを使用して NIC を構成できます。詳細は、[cfgadm\(1M\)](#) のマニュアルページを参照してください。

6. (オプション) リンクの情報を表示します。

`dladm show-phys` コマンドまたは `dladm show-link` コマンドのいずれかを使用して、データリンクに関する情報を表示できます。

例 2-2 新しいネットワークカードを取り付けることによる動的再構成の実行

次の例は、リンク名 `net0` の `bge` カードを `e1000g` カードに交換する方法を示しています。`e1000g` がシステムに接続されると、`net0` のリンク構成が `bge` から `e1000g` に転送されます。

```
# dladm show-phys -L
LINK      DEVICE      LOCATION
```

```
net0    bge0    MB
net1    ibp0    MB/RISER0/PCIE0/PORT1
net2    ibp1    MB/RISER0/PCIE0/PORT2
net3    eoib2   MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

cfgadm コマンドを使用して bge カードを取り外してそこに e1000g カードを取り付けるなど、DR 固有の手順を実行します。カードを取り付けると、e1000g のデータリンクは自動的に名前 net0 を引き継ぎ、そのリンク構成を継承します。

```
# dladm show-phys -L
LINK    DEVICE    LOCATION
net0    e1000g0  MB
net1    ibp0     MB/RISER0/PCIE0/PORT1
net2    ibp1     MB/RISER0/PCIE0/PORT2
net3    eoib2    MB/RISER0/PCIE0/PORT1/cloud-nm2gw-2/1A-ETH-2
```

```
# dladm show-link
LINK    CLASS    MTU    STATE    OVER
net0    phys    9600   up       ---
net1    phys    1500   down    ---
net2    phys    1500   down    --
net3    phys    1500   down    ---
```

▼ SPARC: 各インタフェースの MAC アドレスが一意であることを確認する方法

すべての SPARC ベースのシステムは、システム共通 MAC アドレスを持っており、デフォルトではすべてのインタフェースがこのアドレスを使用します。ただし、アプリケーションによっては、ホスト上のすべてのインタフェースでそれぞれ一意の MAC アドレスが使用されている必要があります。特定のタイプのインタフェース構成 (リンクアグリゲーションや IP ネットワークマルチパス (IPMP) など) では、同様にインタフェースに固有の MAC アドレスが存在する必要があります。

EEPROM パラメータ local-mac-address? によって、SPARC ベースのシステム上のすべてのインタフェースがシステム共通 MAC アドレスまたは一意の MAC アドレスのどちらかを使用しているかを判別します。次の手順では、eeprom コマンドを使用して、local-mac-address? パラメータの現在値をチェックし、必要に応じて変更する方法について説明します。

1. 管理者になります。
2. システム上のすべてのインタフェースがシステム共通 MAC アドレスを現在使用しているかどうかを判断します。

```
# eeprom local-mac-address?  
local-mac-address?=false
```

前の出力では、local-mac-address?=false 設定により、すべてのインタフェースがシステム共通 MAC アドレスを使用することが示されています。たとえば、local-mac-address?=false 設定の値は、インタフェースを IPMP グループのメンバーにする前に、local-mac-address?=true に変更する必要があります。

注記 - リンクアグリゲーションを構成する場合も、この変更を行うべきです。

3. 次のように、local-mac-address? 設定の値を変更します。

```
# eeprom local-mac-address?=true
```

システムをリブートすると、出荷時に設定された MAC アドレスを持つインタフェースは、システム共通 MAC アドレスの代わりに、その出荷時の設定を使用します。出荷時に設定された MAC アドレスを持たないインタフェースは、システム共通 MAC アドレスを引き続き使用します。

4. システム上のすべてのインタフェースの MAC アドレスをチェックします。

複数のインタフェースが 同じ MAC アドレスを持つ場合がないかどうかを調べてください。次の例では、2 つのインタフェースがシステム共通 MAC アドレス 8:0:20:0:0:1 を使用しています。

```
# dladm show-linkprop -p mac-address  
LINK      PROPERTY      PERM VALUE      EFFECTIVE      DEFAULT      POSSIBLE  
net0      mac-address   rw  0:14:4f:f9:b1:a9 0:14:4f:f9:b1:a9 0:14:4f:f9:b1:a9 --  
net3      mac-address   rw  0:14:4f:fb:9a:d4 0:14:4f:fb:9a:d4 0:14:4f:fb:9a:d4 --  
net2      mac-address   rw  0:14:4f:f9:c:d 0:14:4f:f9:c:d 0:14:4f:f9:c:d --  
net1      mac-address   rw  0:14:4f:fa:ea:42 0:14:4f:fa:ea:42 0:14:4f:fa:ea:42 --
```

5. (オプション) すべてのインタフェースが一意の MAC アドレスを持つように、必要に応じて、残りのインタフェースを手動で構成します。

```
# dladm set-linkprop -p mac-address=mac-address interface
```

注記 - この手順が必要なのは、同じ MAC アドレスを持つネットワークインタフェースが複数存在する場合だけです。

前の例では、net0 と net1 をローカルで管理されている MAC アドレスで構成する必要がある場合があります。たとえば、ローカルで管理されている MAC アドレス 06:05:04:03::02 で net0 を再構成するには、次のコマンドを入力します。

```
# dladm set-linkprop -p mac-address=06:05:04:03:02 net0
```

6. システムをリブートします。

◆◆◆ 第 3 章

Oracle Solaris での IP インタフェースとアドレスの構成および管理

この章では、IPv4 および IPv6 アドレス指定を実装するネットワークを構成する方法について説明します。この章のタスクの多くは、IPv4 と IPv6 対応の両方のネットワークに適用されます。IPv4 または IPv6 ネットワークのみに関連する手順については、そのことが示されます。

ネットワークを構成する前に、『Oracle Solaris 11.2 でのネットワーク配備の計画』に説明されている IP 関連の計画タスクを確認します。

グローバルパケット転送やトランスポートレイヤーサービスなど、ほかの TCP/IP プロパティの管理については、『Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理』の「トランスポート層サービスの管理」を参照してください。

この章の内容は、次のとおりです。

- 50 ページの「[ipadm コマンドを使用したネットワーク構成の管理](#)」
- 50 ページの「[IPv4 インタフェースの構成](#)」
- 56 ページの「[IPv6 インタフェースの構成](#)」
- 65 ページの「[IPv4 ネットワークから IPv6 ネットワークへの移行](#)」
- 66 ページの「[ルーティングの構成](#)」
- 75 ページの「[マルチホームホストの構成](#)」
- 79 ページの「[IP インタフェースプロパティとアドレスのカスタマイズ](#)」
- 81 ページの「[IP アドレスプロパティのカスタマイズ](#)」
- 82 ページの「[IP インタフェース構成の無効化、削除、および変更](#)」
- 85 ページの「[IP インタフェースとアドレスのモニタリング](#)」

ipadm コマンドを使用したネットワーク構成の管理

ipadm コマンドは、IP インタフェースを構成する主な手段としての ifconfig コマンドを最終的に置き換えるために導入されました。

ipadm コマンドは、次の TCP/IP プロトコルのプロパティを構成するための ndd コマンドも置き替えます。

- IP
- アドレス解決プロトコル (ARP)
- ストリーム制御伝送プロトコル (SCTP)
- インターネット制御メッセージプロトコル (ICMP)
- TCP およびユーザーデータグラムプロトコル (UDP) などの上位階層プロトコル

インタフェースを構成するためのツールとして、ipadm コマンドには ifconfig コマンドに優る次の利点があります。

- ifconfig コマンドの提供する構造よりも優れたオブジェクト指向サブコマンド構造を提供します。この変更により、最終的にネットワーク構成手順がよりわかりやすくなります。
- ifconfig コマンドとは異なり、ネットワーク構成の変更を永続的なものにできます。
- スクリプト作成に役立つ解析可能な出力オプションをサポートします。

プロトコルのプロパティを設定するためのツールとして、ipadm コマンドには ndd に優る次の利点があります。

- プロパティの現在値とデフォルト値や指定可能な値範囲などの、ndd コマンドよりも広範囲のプロパティ情報を提供します。
- プロパティ値を永続的 (または一時的) に設定します。ndd コマンドは、プロパティ値を一時的に設定するだけです。
- スクリプト作成に役立つ解析可能な出力オプションをサポートします。

ipadm コマンドを ifconfig および ndd コマンドと比較する場合は、『[Oracle Solaris 10 から Oracle Solaris 11.2 への移行](#)』の「[ネットワーク管理コマンドの変更](#)」を参照してください。

ipadm コマンドの詳細については、[ipadm\(1M\)](#) のマニュアルページを参照してください。

IPv4 インタフェースの構成

次の手順および例では、IPv4 アドレスを使用するネットワークを構成する方法について説明します。

▼ IPv4 インタフェースを構成する方法

始める前に システム上でどの NCP がアクティブであるかを確認して、構成が正しいプロファイルに適用されていることを確認します。[例6-6「固定モードとリアクティブモードの切り替え」](#)を参照してください。

1. 管理者になります。
2. インタフェースを作成します。

```
# ipadm create-interface-class interface
```

interface-class ユーザーが作成できる次の 3 つのインタフェースクラスの中の 1 つを示します。

■ IP インタフェース

このインタフェースクラスは、ネットワーク構成を実行するときに作成するもっとも一般的なクラスです。このインタフェースクラスを作成するには、`create-ip` サブコマンドを使用します。

■ STREAMS 仮想ネットワークインタフェース (VNI インタフェース)

このインタフェースクラスを作成するには、`create-vni` サブコマンドを使用します。

Oracle Solaris 11.2 以降、VNI インタフェースにより自由に名前を付けることができます。以前は、VNI インタフェース名には、`vni0` のように接頭辞に「vni」を含める必要がありました。この要件は適用されなくなりました。VNI デバイスとインタフェースの詳細は、[vni\(7d\)](#) および [ipadm\(1M\)](#) のマニュアルページを参照してください。

■ IPMP インタフェース

このインタフェースクラスは、IPMP グループを構成するときに使用されます。このインタフェースクラスを作成するには、`create-ipmp` サブコマンドを使用します。IPMP グループの詳細は、『[Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理](#)』の第 2 章「IPMP の管理について」を参照してください。

interface インタフェースの名前を示します。この名前は、インタフェースの作成に使用しているデータリンクの名前と同じです。システム上にあるデータリンクを表示するには、`dladm show-link` コマンドを使用します。

3. 次のいずれかのコマンドを使用して、有効な IP アドレスを持つ IP インタフェースを構成します。

- 静的 IP アドレスを構成します。

```
# ipadm create-addr -a address [interface | addrobj]
```

`-a address` インタフェース上に構成する IP アドレスを指定します。

注記 - トンネル構成では、通常、トンネルインタフェースにローカルアドレスとリモートアドレスの 2 つのアドレスが必要です。ローカルとリモートのアドレスおよびトンネル構成については、『Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理』の第 5 章「IP トンネルの管理」を参照してください。

数値 IP アドレスの場合は、CIDR (Classless Inter-Domain Routing) 表記法を使用します。CIDR 表記法を使用しない場合、ネットマスクは `svc:/system/name-service/switch:default` ネットマスクデータベース検索順序またはクラスフルアドレスのセマンティクスを使用して決定されます。

必要に応じて、数値 IP アドレスの代わりにホスト名を指定できます。ホスト名の使用が有効なのは、そのホスト名に対応する数値 IP アドレスが `/etc/hosts` ファイルで定義されている場合です。このファイルで数値 IP アドレスが定義されていない場合は、`name-service/switch` サービスで `host` に対して指定されているリゾルバの順序を使用して、この数値が一意に取得されます。特定のホスト名に対して複数のエントリが存在する場合は、エラーが生成されます。

注記 - ブートプロセス中、IP アドレスはネームサービスがオンラインになる前に構成されます。そのため、ネットワーク構成で使用されているすべてのホスト名が `/etc/hosts` ファイルで定義されていることを確認する必要があります。

[*interface* | *addrobj*] Oracle Solaris では、各アドレスは対応するアドレスオブジェクトによって識別され、コマンド内で *addrobj* によって表されます。そのアドレスに対するその後の構成では、実際の IP アドレスではなくアド

レスオブジェクトを参照します。たとえば、`ipadm show-addr addrobj` や `ipadm delete-addr addrobj` のように入力します。アドレスオブジェクト名を自動的に生成するには、*interface* にインタフェース名を指定します。アドレスオブジェクトに手動で名前を付けるには、アドレスオブジェクト名を直接指定します。

- インタフェース名を指定すると、アドレスオブジェクトに *interface/address-family* という形式の名前が自動的に付けられます。*Address family* は、IPv4 アドレスを示す *v4* または IPv6 アドレスを示す *v6* のいずれかです。自動生成されたアドレスオブジェクト名を使用して、複数のアドレスがインタフェース上で構成される場合、一意の名前になるよう、アルファベット文字がアドレスオブジェクト名に追加されます。たとえば、`net0/v4`、`net0/v4a`、`net0/v4b`、`net0/v6`、`net0/v6a` などのようになります。
- *addrobj* に手動でアドレスオブジェクトを指定する場合は、*interface/user-specified-string* という形式を使用する必要があります。*user-specified-string* は、アルファベット文字で始まり、最大長が 32 文字の英数字の文字列を示します。たとえば、`net0/static`、`net0/static1`、`net1/private` などのアドレスオブジェクトを指定できます。

- 静的でないアドレスを構成します。

```
# ipadm create-addr -T address-type [interface | addrobj]
```

ここで、*address-type* は `dhcp` または `addrconf` のいずれかです。`addrconf` 引数は、自動的に生成される IPv6 アドレスを指します。

interface および *addrobj* オプションの詳細は、静的アドレスの作成に関する前の説明を参照してください。

4. (オプション) 新しく構成された IP インタフェースに関する情報を表示します。

調べる対象の情報に応じて、次のコマンドを使用できます。

```
# ipadm interface
```

サブコマンドを指定しない場合は、システム上のすべてのインタフェースの情報が表示されます。

```
# ipadm show-if interface
```

interface を指定しない場合は、システム上のすべてのインタフェースの情報が表示されます。

```
# ipadm show-addr interface|addrobj
```

interface または *addrobj* を指定しない場合は、すべてのアドレスオブジェクトの情報が表示されます。

`ipadm show-*` サブコマンドの出力についての詳細は、[85 ページの「IP インタフェースとアドレスのモニタリング」](#)を参照してください。

5. ホスト名を使用する静的 IP アドレスを構成している場合は、`/etc/hosts` ファイルにその IP アドレスのエントリを追加します。

このファイル内のエントリは、IP アドレスとそれに対応するホスト名で構成されています。

注記 - DHCP アドレスを構成している場合は、`/etc/hosts` ファイルを更新する必要はありません。

6. デフォルトルートを定義します。

```
# route -p add default address
```

ルーティングテーブルの内容を確認するには、`netstat -r` コマンドを使用します。

ルートの管理の詳細は、[route\(1M\)](#) のマニュアルページおよび [68 ページの「永続的 \(静的\) ルートの作成」](#)を参照してください。

例 3-1 静的 IP アドレスを使用した IPv4 インタフェースの構成

次の例では、静的 IP アドレスを使用してインタフェースを構成する方法を示します。この例では、最初にシステム上で `DefaultFixed NCP` を有効化して、`dladm` および `ipadm` コマンドによりリアクティブ NCP が変更されないようにします (環境によっては実行する手動ネットワーク構成が無効にされる可能性がある)。

```
# netadm enable -p ncp DefaultFixed

# dladm show-phys
LINK      MEDIA      STATE      SPEED      DUPLEX      DEVICE
net3      Ethernet   up         100Mb     full        bge3

# dladm show-link
LINK      CLASS      MTU        STATE      OVER
net3      phys       1500       up         --         --

# ipadm create-ip net3
# ipadm create-addr -a 192.168.84.3/24 net3
net3/v4
```

```
# ipadm
NAME      CLASS/TYPE  STATE  UNDER  ADDR
lo0       loopback   ok     --     --
lo0/v4    static     ok     --     127.0.0.1/8
lo0/v6    static     ok     --     ::1/128
net3      ip         ok     --     --
net3/v4   static     ok     --     192.168.84.3/24

# vi /etc/hosts
# Internet host table
# 127.0.0.1    localhost
10.0.0.14    myhost
192.168.84.3 sales1

# route -p add default 192.168.84.1
# netstat -r
Routing Table: IPv4
Destination      Gateway          Flags Ref  Use      Interface
-----
default          192.168.84.1    UG    2    10466
192.168.84.0    192.168.84.3    U     3    1810    net0
localhost       localhost       UH    2     12     lo0

Routing Table: IPv6
Destination/Mask  Gateway          Flags Ref  Use  If
-----
solaris          solaris          UH    2    156  lo0
```

sales1 が /etc/hosts ファイル内で定義されている場合は、次のアドレスを割り当てるときにそのホスト名を使用できます。

```
# ipadm create-addr -a sales1 net3
net3/v4
```

例 3-2 DHCP サーバーから IP アドレスを受信するネットワークインタフェースの構成

次の例では、アドレスを DHCP サーバーから受け取るように IP インタフェースを構成します。通常、DHCP はデフォルトルートもインストールします。そのため、この例にはデフォルトルートを手動で追加する手順も含まれます。

```
# dladm show-phys
LINK      MEDIA      STATE  SPEED  DUPLEX  DEVICE
net3      Ethernet   up     100Mb  full    bge3

# dladm show-link
LINK      CLASS  MTU  STATE  OVER
net3     phys   1500 up     --    --

# ipadm create-ip net3
# ipadm create-addr -T dhcp net3
net3/v4
```

```
# ipadm
NAME      CLASS/TYPE  STATE  UNDER  ADDR
lo0       loopback    ok     --      --
lo/v4     static      ok     --      127.0.0.1/8
net3      ip          ok     --      --
net3/v4   dhcp        ok     --      10.0.1.13/24
```

IPv6 インタフェースの構成

ネットワーク上で IPv6 アドレス指定を使用する最初の手順として、システムの IP インタフェース上で IPv6 を構成する必要があります。インストール時に、1 つまたは複数のインタフェース上で IPv6 を有効にできます。

インストール時に IPv6 サポートを有効にした場合、インストール完了後に次の IPv6 関連のファイルやテーブルが存在しています。

- IPv6 アドレスを使用した検索が行えるように、`name-service/switch` SMF サービスが変更されます。
- IPv6 アドレス選択ポリシーテーブルが作成されます。このテーブルは、IPv6 が有効なインタフェース経由の転送で使用される IP アドレス書式に優先順位を付けます。

▼ IPv6 用にシステムを構成する方法

次の手順では、Oracle Solaris のインストール後に追加されたインタフェースで IPv6 を有効にする方法について説明します。IPv6 を構成する手順は、IPv6 ノードになるすべてのシステムインタフェースで IPv6 を有効にすることから始まります。通常の IPv6 配備では、自動構成を使用して IP インタフェースを構成します。`autoconf` IP アドレスは、リンクローカルアドレスを割り当て、サブネット上で使用中の接頭辞およびルーターを検出します。それらのノードの構成は、IPv6 ネットワーク上の機能 (ホスト、サーバー、またはルーター) に基づいて調整できます。`autoconf` に対して設定されるインタフェースは、DHCPv6 アドレス情報を自動的に要求することも行います。自動構成や DHCPv6 なしで静的 IPv6 アドレスのみを有効にするには、`ipadm` コマンドと適切なオプションを使用して、動的に割り当てられるほかのアドレスを一切追加せずに、インタフェース上でリンクローカルアドレスを作成します。例については、[65 ページの「IPv4 ネットワークから IPv6 ネットワークへの移行」](#)を参照してください。

注記 - インタフェースと同じリンク上に IPv6 接頭辞を現在通知しているルーターが存在する場合、そのインタフェースは自動構成アドレスの一部としてそのサイトの接頭辞を取得します。詳細は、『[ルーターまたはロードバランサとしての Oracle Solaris 11.2 システムの構成](#)』の「[IPv6 対応のルーターを構成する方法](#)」を参照してください。

1. (オプション) `ipadm create-ip` コマンドと適切なオプションを使用して、IP インタフェースを構成します。

```
# ipadm create-ip interface
```

たとえば、次のようにして `net0` 用の IP インタフェースを構成します。

```
# ipadm create-ip net0
```

IPv4 で使用するインタフェースが構成済みである場合、この手順は必要ありません。IP インタフェースの一般的な構成手順については、[IPv4 インタフェースを構成する方法](#)を参照してください。

2. 1 つまたは複数の IP アドレスを割り当てます。

注記 - IP アドレスを割り当てるときには、必ず次のように正しいオプションを使用して IPv6 アドレスを割り当てます。

```
# ipadm create-addr -T addrconf interface
```

アドレスをさらに追加する場合は、次の構文を使用します。

```
# ipadm create-addr -a ipv6-address interface
```

3. (オプション) 静的 IPv6 デフォルトルートを作成します。

```
# /usr/sbin/route -p add -inet6 default ipv6-address
```

注記 - 自動構成の一部として、`in.ndpd` はデフォルトルートをその検出時に追加するため、手動構成されたデフォルトルートを含む複数のデフォルトルートが使用可能になる可能性があります。システムは、使用可能なすべてのルートに基づいてデフォルトルートを選択するため、手動構成されたデフォルトルートが常に使用されるとは限りません。

4. (オプション) ノード上にあるインタフェース変数のパラメータを定義する `/etc/inet/ndpd.conf` ファイルを作成します。

ホストのインタフェースに一時アドレスを作成する必要がある場合は、59 ページの「IPv6 インタフェースに対する一時アドレスの使用」を参照してください。/etc/inet/ndpd.conf の詳細については、[ndpd.conf\(4\)](#) のマニュアルページを参照してください。

5. (オプション) 次のようにして、IP インタフェースのステータスおよびその IPv6 構成を表示します。

```
# ipadm show-addr
```

例 3-3 インストール後に IPv6 インタフェースを有効にする方法

次の例では、net0 インタフェースの IPv6 を有効にする方法を示します。作業を始める前に、システムに構成されているすべてのインタフェースのステータスを確認します。

```
# ipadm show-addr
ADDROBJ  TYPE  STATE  ADDR
lo0/v4   static ok     127.0.0.1/8
net0/v4   static ok     172.16.27.74/24
```

前の出力に示されているように、このシステムでは現在 net0 インタフェースのみが構成されています。net0 インタフェースが未構成の場合は、ipadm create-ip net0 コマンドを使用してインタフェースを稼働させます。

次のようにして、このインタフェース上で IPv6 を有効にします。

```
# ipadm create-addr -T addrconf net0
# ipadm create-addr -a 2001:db8:3c4d:15::203/64 net0

# ipadm show-addr
ADDROBJ  TYPE  STATE  ADDR
lo0/v4   static ok     127.0.0.1/8
net0/v4   static ok     172.16.27.74/24
net0/v6   addrconf ok     fe80::203:baff:fe13:14e1/10
lo0/v6   static ok     ::1/128
net0/v6a  static ok     2001:db8:3c4d:15::203/64

# route -p add -inet6 default fe80::203:baff:fe13:14e1
```

- 次の手順
- IPv6 ノードをルーターとして構成する方法については、『[ルーターまたはロードバランサとしての Oracle Solaris 11.2 システムの構成](#)』の「IPv6 ルーターの構成」を参照してください。
 - ノードをサーバーとして調整する方法については、64 ページの「[サーバー上での IPv6 が有効なインタフェースの構成](#)」を参照してください。

IPv6 インタフェースに対する一時アドレスの使用

IPv6 一時アドレスには、インタフェースの MAC アドレスの代わりに、インタフェース ID としてランダムに生成された 64 ビットの数字が含まれます。匿名にしておきたい IPv6 ノード上の任意のインタフェースに対しては、一時アドレスを使用できます。たとえば、公開 Web サーバーにアクセスする必要があるホストのインタフェースに対しては、一時アドレスを使用したい場合があります。一時アドレスには、IPv6 プライバシー拡張が実装されます。これらの拡張機能については、RFC 3041 “[Privacy Extensions for Stateless Address Autoconfiguration in IPv6](http://www.rfc-editor.org/rfc/rfc3041.txt)” (<http://www.rfc-editor.org/rfc/rfc3041.txt>) を参照してください。

1 つまたは複数のインタフェースに対して一時アドレスを有効にする必要がある場合は、`/etc/inet/ndpd.conf` ファイルを使用します。しかし、標準の自動構成された IPv6 アドレスとは異なり、一時アドレスは、64 ビットのサブネット接頭辞とランダムに生成された 64 ビット数から構成されます。このランダムな数は、IPv6 アドレスのインタフェース ID 部分になります。リンクローカルアドレスでは、一時アドレスはインタフェース ID としては生成されません。

一時アドレスの *preferred lifetime* のデフォルトは、1 日です。一時アドレスの生成を有効にした場合、`/etc/inet/ndpd.conf` ファイルでは次の変数も構成できます。

<i>valid lifetime</i> TmpValidLifetime	一時アドレスが存在できる寿命。この寿命を過ぎると、そのアドレスはホストから削除されます。
<i>preferred lifetime</i> TmpPreferredLifetime	一時アドレスが無効にされるまでの時間。この時間は、 <i>valid lifetime</i> よりも短くします。
<i>address regeneration</i>	<i>preferred lifetime</i> が満了するまでの時間。この時間内に、ホストは新しい一時アドレスを生成します。

一時アドレスの時間を表現するには、次の書式を使用します。

<i>n</i>	<i>n</i> 秒数 (デフォルト)
<i>n h</i>	<i>n</i> 時間数 (h)
<i>n d</i>	<i>n</i> 日数 (d)

▼ 一時 IPv6 アドレスを構成する方法

1. 必要に応じて、ホストのインタフェースの IPv6 を有効にします。

56 ページの「IPv6 用にシステムを構成する方法」を参照してください。

2. `/etc/inet/ndpd.conf` ファイルを編集して、一時アドレスの生成を有効にします。

- ホストのすべてのインタフェースに対して一時アドレスを構成するには、次の行を `/etc/inet/ndpd.conf` ファイルに追加します。

```
ifdefault TmpAddrsEnabled true
```

- 特定のインタフェースに対して一時アドレスを構成するには、次の行を `/etc/inet/ndpd.conf` ファイルに追加します。

```
if interface TmpAddrsEnabled true
```

3. (オプション) 一時アドレスの `valid lifetime` を指定します。

```
ifdefault TmpValidLifetime duration
```

この構文は、ホストのすべてのインタフェースに対して `valid lifetime` を指定します。`duration` の値は、秒、時間、または日です。`valid lifetime` のデフォルトは 7 日です。`TmpValidLifetime` に `if interface` キーワードを使用すると、特定のインタフェースに対して一時アドレスの `valid lifetime` を指定できます。

4. (オプション) 一時アドレスの `preferred lifetime` を指定します。この寿命を過ぎると、一時アドレスは非推奨になります。

```
if interface TmpPreferredLifetime duration
```

この構文は、特定のインタフェースに対して一時アドレスの `preferred lifetime` を指定します。`preferred lifetime` のデフォルトは 1 日です。`TmpPreferredLifetime` に `ifdefault` キーワードを使用すると、ホストのすべてのインタフェースに対して一時アドレスの `preferred lifetime` を指定できます。

注記 - デフォルトアドレス選択では、無効にされた IPv6 アドレスには低い優先順位が与えられます。IPv6 一時アドレスが非推奨になると、デフォルトアドレス選択によって、パケットのソースアドレスとして非推奨でないアドレスが選択されます。非推奨でないアドレスは、自動的に生成された IPv6 アドレス、またはインタフェースの IPv4 アドレス (可能な場合) になります。デフォルトアドレス選択の詳細については、『[Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理](#)』の「[デフォルトアドレス選択の管理](#)」を参照してください。

5. (オプション) アドレスを無効にするまでの時間を指定します。この間に、ホストは新しい一時アドレスを生成する必要があります。

```
ifdefault TmpRegenAdvance duration
```

この構文は、ホストのすべてのインタフェースに対して、一時アドレスを無効にするまでの時間を指定します。デフォルトは 5 秒です。

6. 次のように、`in.ndpd` デーモンの構成を変更します。

```
# pkill -HUP in.ndpd
# /usr/lib/inet/in.ndpd
```

7. **Example 3-4** で示すように、**例3-4「一時アドレスを有効にした状態での ipadm show-addr コマンド出力の表示」** コマンドを発行して一時アドレスが作成されたことを確認します。コマンド出力では、一時アドレスの `CURRENT` フィールドに `t` フラグが表示されます。

例 3-4 一時アドレスを有効にした状態での `ipadm show-addr` コマンド出力の表示

次に、一時アドレスを作成したあとの `ipadm show-addr` コマンドの出力の例を示します。このサンプル出力には IPv6 関連の情報のみが含まれています。

```
# ipadm show-addr -o all
ADDROBJ  TYPE      STATE CURRENT PERSISTENT ADDR
lo0/v6   static   ok    U----   ---      ::1/128
net0/v6   addrconf ok    U----   ---      fe80::a00:20ff:feb9:4c54/10
net0/v6a  static   ok    U----   ---      2001:db8:3c4d:15:a00:20ff:feb9:4c54/64
net0/?    addrconf ok    U--t-   ---      2001:db8:3c4d:15:7c37:e7d1:fc9c:d2cb/64
```

アドレスオブジェクト `net0/?` では、`t` フラグが `CURRENT` フィールドの下に設定されており、対応するアドレスが一時インタフェース ID を保持していることを示しています。

- 参照**
- IPv6 アドレス用のネームサービスサポートを設定する場合は、[第4章「Oracle Solaris クライアントでのネームサービスとディレクトリサービスの管理」](#)を参照してください。
 - サーバー上で IPv6 アドレスを構成する方法については、[62 ページの「ユーザー指定の IPv6 トークンを構成する方法」](#)を参照してください。
 - IPv6 ノード上のアクティビティをモニターする場合は、『[Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理](#)』の第 1 章「TCP/IP ネットワークの管理」を参照してください。

IPv6 トークンの構成

IPv6 アドレスの 64 ビットのインタフェース ID は、トークンとも呼ばれます。トークンは、アドレスが自動構成されるときに、インタフェースの MAC アドレスに関連付けられます。ほとんどの場

合、ルーティングを行わないノード (IPv6 ホストと IPv6 サーバー) では、自動構成されたトークンを使用するようにしてください。

ただし、システムが保守されるときにインタフェースが定期的に交換されるサーバーでは、自動構成されたトークンを使用すると問題が発生することがあります。インタフェースカードが変更されると、MAC アドレスも変更されます。その結果、IP アドレスが変わらないことを前提とするサーバーでは、問題が発生することがあります。ドメインネームシステム (DNS) やネットワーク情報システム (NIS) など、ネットワークインフラストラクチャーのさまざまな部分に、サーバーのインタフェース用の特定の IPv6 アドレスが格納されている場合があります。

アドレスが変わることで発生する問題を回避するために、IPv6 アドレスのインタフェース ID として使用されるトークンを手動で構成できます。トークンを作成するには、IPv6 アドレスのインタフェース ID 部分に相当する 64 ビット以下の 16 進数を指定します。それ以降は、アドレスが自動構成されるときに近傍検索によって作成されるインタフェース ID は、インタフェースの MAC アドレスからは作成されません。代わりに、手動で作成したトークンがインタフェース ID になります。このトークンは、カードを交換しても、インタフェースに割り当てられたままになります。

注記 - ユーザー指定のトークンと一時アドレスとの違いは、一時アドレスがランダムに生成されるのに対し、ユーザー指定のトークンはユーザーが明示的に作成する点です。

▼ ユーザー指定の IPv6 トークンを構成する方法

次の手順は、インタフェースが定期的に置き換えられるサーバーで特に役立ちます。また、これらの手順に従って、任意の IPv6 ノード上でユーザー固有のトークンを構成することもできます。

1. トークンの構成対象となるインタフェースが存在しており、かつそのインタフェースで IPv6 アドレスが 1 つも構成されていないことを確認します。

```
# ipadm show-if
IFNAME  CLASS      STATE  ACTIVE  OVER
lo0     loopback  ok     yes     ---
net0    ip         ok     yes     ---

# ipadm show-addr
ADDROBJ  TYPE      STATE  ADDR
lo0/v4   static   ok     127.0.0.1/8
```

前の出力は、ネットワークインタフェース `net0` が存在しており、IPv6 アドレスは一切構成されていないことを示しています。

2. ノードのインタフェースのトークンとして使用する、次の形式の 1 つまたは複数の 64 ビットの 16 進数を作成します。

```
XXXX.XXXX.XXXX.XXXX
```

3. ユーザー固有のインタフェース ID (トークン) を保持する各インタフェースを構成します。

```
# ipadm create-addr -T addrconf -i interface-ID interface
```

たとえば、次のように、トークンを使用してインタフェース `net0` を構成します。

```
# ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0
```

注記 - トークンを使用してアドレスオブジェクトが作成されると、そのトークンはもう変更できなくなります。

4. 変更に合わせて、IPv6 デーモンを更新します。

```
# pkill -HUP in.ndpd
```

例 3-5 ユーザー指定のトークンを IPv6 インタフェースに構成する

次の例では、IPv6 アドレスとトークンを使用して `net0` を構成する方法を示します。

```
# ipadm show-if
IFNAME  CLASS    STATE   ACTIVE  OVER
lo0     loopback ok       yes     ---
net0    ip       ok       yes     ---

# ipadm show-addr
ADDROBJ  TYPE      STATE  ADDR
lo0/v4   static   ok     127.0.0.1/8

# ipadm create-addr -T addrconf -i ::1a:2b:3c:4d/64 net0
# pkill -HUP in.ndpd
# ipadm show-addr
ADDROBJ  TYPE      STATE  ADDR
lo0/v6   static   ok     ::1/128
net0/v6  addrconf ok     fe80::1a:2b:3c:4d/10
net0/v6a addrconf ok     2002:a08:39f0:1:1a:2b:3c:4d/64
```

トークンの構成が完了したあと、アドレスオブジェクト `net0/v6` は、`1a:2b:3c:4d` がインタフェース ID として構成されたリンクローカルアドレスと別のアドレスの両方を持っています。`net0/v6` が作成されたあとでは、このインタフェースのトークンはもう変更できません。

参照 ■ サーバーの IPv6 アドレスを使用してネームサービスを更新する場合は、[第4章「Oracle Solaris クライアントでのネームサービスとディレクトリサービスの管理」](#)を参照してください。

- サーバーのパフォーマンスをモニターする場合は、『Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理』の第 1 章「TCP/IP ネットワークの管理」を参照してください。

サーバー上での IPv6 が有効なインタフェースの構成

サーバーで IPv6 アドレスを使用することを計画するときは、サーバーのインタフェースの IPv6 を有効にするために、いくつかのことを決定する必要があります。それらの決定は、インタフェースの IPv6 アドレスのインタフェース ID (トークン) を構成するときに、どのような方法を採用するかに影響します。

▼ サーバーのインタフェースの IPv6 を有効にする方法

次の手順では、ネットワークのサーバー上で IPv6 を有効にする方法を示します。IPv6 の実装方法によっては一部の手順が変わる可能性があります。

1. **サーバーの IP インタフェースで IPv6 を有効にします。**
手順については、56 ページの「IPv6 インタフェースの構成」を参照してください。
2. **サーバーと同じリンク上のルーターに IPv6 サブネット接頭辞が構成されていることを確認します。**
詳細は、『ルーターまたはロードバランサとしての Oracle Solaris 11.2 システムの構成』の「IPv6 ルーターの構成」を参照してください。
3. **インタフェース ID をサーバーの IPv6 が有効なインタフェースに割り当てるため、次の戦略のいずれかから選択します。**
デフォルトでは、IPv6 アドレスの自動構成によって IPv6 アドレスのインタフェース ID 部分が作成されるときに、インタフェースの MAC アドレスが使用されます。インタフェースの IPv6 アドレスが既知の場合には、インタフェースが切り替わると、問題が発生することがあります。新しいインタフェースの MAC アドレスは、別のアドレスになります。アドレスが自動構成されると、新しいインタフェース ID が生成されます。
 - IPv6 が有効なインタフェースを置き換える予定がない場合は、自動構成された IPv6 アドレスを使用します。

- IPv6 が有効なインタフェースをローカルネットワークの外部には匿名で表示する必要がある場合は、ランダムに生成されたトークンをインタフェース ID に使用することを検討します。

詳細は、59 ページの「一時 IPv6 アドレスを構成する方法」を参照してください

- IPv6 が有効なインタフェースを定期的に切り替えて使用する場合は、静的な構成を使用するか、インタフェース ID のトークンを作成できます。

詳細は、62 ページの「ユーザー指定の IPv6 トークンを構成する方法」を参照してください。

IPv4 ネットワークから IPv6 ネットワークへの移行

注記 - IPv4 ネットワークから IPv6 ネットワークへ移行する前に、大半の移行計画では、IPv4 と IPv6 を同時に長期間、場合によっては無期限で実行しなければならないことに留意してください。

IPv4 ネットワークから IPv6 ネットワークに移行する前に、『Oracle Solaris 11.2 でのネットワーク配備の計画』の第 2 章「IPv6 アドレスの使用の計画」の情報を確認して、追加タスクの実行が必要かどうかを判断してください。

IPv4 ネットワークから IPv6 ネットワークへの移行の基本手順には、最初にすべての既存の IPv4 DHCP および静的 IP アドレスを削除してから、新しい IPv6 アドレスを必要な数だけ再構成することが含まれます。新しい IPv6 インタフェースが、現在 IPv6 接頭辞を通知しているルーターと同じリンク上にある場合は、インタフェースはそのリンク接頭辞を取得します。詳細は、『ルーターまたはロードバランサとしての Oracle Solaris 11.2 システムの構成』の「IPv6 ルーターの構成」を参照してください。

例 3-6 IPv4 アドレスから IPv6 アドレスへの移行

次の例は、既存の IPv4 アドレスから IPv6 アドレスに移行する方法を示します。このプロセスは、既存の IPv4 DHCP および静的 IP アドレスをすべて削除することから始まります。

```
# ipadm show-addr net0/
ADDROBJ  TYPE  STATE  ADDR
lo0/v4   static ok     127.0.0.1/8
net0/v4  static ok     172.16.27.74/24
# ipadm delete-addr net0/v4
```

手順については、[84 ページの「IP インタフェース構成の削除または変更」](#)を参照してください。

次に、`ipadm create addr` コマンドおよび適切なオプションと引数を使用して、新しい IPv6 アドレスが作成されます。

たとえば、次のようにリンクローカルおよび `addrconf` IPv6 アドレスを作成できます。

```
# ipadm create-addr -T addrconf -p stateless=yes,stateful=yes net0/v6a
```

DHCPv6 なしの静的 IPv6 アドレスおよび `addrconf` アドレスを作成します。

```
# ipadm create-addr -T addrconf -p stateless=no,stateful=no net0/v6a
# ipadm create-addr -T static -a a::b/64 net0/v6b
```

次のようにして、静的 IPv6 アドレスを作成します。

```
# ipadm create-addr -T static -a a::b/64 net0/v6b
```

`ipadm show-addr` コマンドを使用して、新しい IPv6 構成を表示します。

この例に含まれない追加の IPv6 構成手順 (必須およびオプション) については、[56 ページの「IPv6 インタフェースの構成」](#)を参照してください。

ルーティングの構成

このセクションには、次のトピックが含まれます。

- [66 ページの「ルーティングテーブルとルーティングの種類」](#)
- [68 ページの「永続的 \(静的\) ルートの作成」](#)
- [71 ページの「単一インタフェースシステムのルーティングの有効化」](#)

ルーティングテーブルとルーティングの種類

ルーターとホストの両方がルーティングテーブルを保守します。たとえば、次のルーティングテーブルには、システムのデフォルトのローカルネットワークも含め、システムで知られているネットワークの IP アドレスがリストされています。このテーブルには、既知の各ネットワークに対するゲートウェイシステムの IP アドレスもリストされています。「ゲートウェイ」とは、発信されるパケットを受け取り、ローカルネットワークより 1 ホップ外側に転送するシステムのことです。

```

Routing Table: IPv4
Destination      Gateway          Flags Ref  Use  Interface
-----
default          172.20.1.10     UG    1   532  net0
224.0.0.0        10.0.5.100     U     1     0  net1
10.0.0.0          10.0.5.100     U     1     0  net1
127.0.0.1         127.0.0.1      UH    1     57  lo0

```

Oracle Solaris システムでは、静的および動的という 2 種類のルーティングを構成できます。1 つのシステムに、これらの経路制御のどちらか一方を構成することも、両方を構成することもできます。動的ルーティングを実装するシステムは、IPv4 ネットワークの場合はルーティング情報プロトコル (RIP)、IPv6 ネットワークの場合は RIPng (次世代 RIP) などのルーティングプロトコルを利用して、ネットワークトラフィックをルーティングし、テーブル内のルーティング情報を更新します。静的ルーティングの場合、情報は `route` コマンドを使用して手動で維持されます。詳細は、[route\(1M\)](#) のマニュアルページを参照してください。

ローカルネットワークまたは自律システム (AS) のルーティングを構成するときは、特定のルーターやホストでどの種類のルーティングをサポートするかを検討してください。次の表に、ルーティングのタイプと、各ルーティングタイプを適用するのにもっとも適したネットワークのシナリオを示します。

ルーティングタイプ	最適な使用
静的	デフォルトルーターからルートを取得する小規模なネットワークとホスト、および隣接する数ホップの範囲にある 1 つか 2 つのルーターに関する情報のみを必要とするデフォルトルーター。
動的	多数のホストが存在するローカルネットワーク上のルーター、および大規模な自律システム上のホストを含む、より規模の大きいインターネットワーク。動的ルーティングは、ほとんどのネットワークのシステムで最適なオプションです。
静的経路制御と動的経路制御の組み合わせ	静的にルーティングされるネットワークと動的にルーティングされるネットワークを接続するルーター、および、内部の自律システムと外部のネットワークを接続するボーダールーター。1 つのシステムで静的ルーティングと動的ルーティングの両方を組み合わせることは、一般的に行われています。

『Oracle Solaris 11.2 でのネットワーク配備の計画』の「IPv4 自律システムのトポロジ」で説明されているトポロジは、静的ルーティングと動的ルーティングを組み合わせたものです。

注記 - 同じ宛先へのルートが 2 つあっても、システムで負荷分散やフェイルオーバーが自動的に行われるわけではありません。これらの機能が必要な場合は、IPMP を使用します。詳細は、『Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理』の第 2 章「IPMP の管理について」を参照してください。

永続的 (静的) ルートの作成

route コマンドを使用して手動でネットワークルーティングテーブルを操作します。変更をリブート後も永続させるには、-p オプションを使用します。/etc/defaultrouter ファイルは Oracle Solaris 11 では非推奨であるため、このファイルを使用してルート (デフォルトまたはそれ以外) を管理できなくなりました。route コマンドの使用は、手動でリブート後もルートを永続させることのできる唯一の方法です。

注記 - route コマンドは、アクティブなプロファイルのルートのみを操作します。デフォルトルート、およびほかのすべてのルートは、アクティブなプロファイルが変更された場合に置き換えられる可能性があります。ただし、システムで同じプロファイルを常に使用する場合、これは関係ありません。

ルートを永続的に追加する場合は、追加するルートが永続的構成にすでに存在しないことを確認する必要があります。これらのルートが永続的構成内に存在する場合、ネットワークルーティングテーブルは永続的ルートを更新せずに変化する可能性があります。1 つの例は、システムのデフォルトルートがシステムのプライマリインタフェースにマップされる状況です (Oracle Solaris インストール後によくあるケースです)。その後、システムのプライマリインタフェースを別のインタフェースに変更すると、システムのデフォルトルートも永続的に変更されます。ベストプラクティスは、新しいルートを追加する前に、永続的ルート構成を削除することです。詳細は、『Oracle Solaris 11.2 でのネットワーク管理のトラブルシューティング』の「永続的なルートを追加するときの問題のトラブルシューティング」を参照してください。

永続的ルートの作成および表示に関する、次の追加情報を確認してください。

- route コマンドと -p オプションを使用して、ルートを永続的に追加します。

```
# route -p add default ip-address
```

この方法を使用して作成されるルートの場合、route - p show コマンドを使用して、永続的な静的ルートをすべて表示します。

```
# route -p show
```

- netstat コマンドと次のオプションを使用して、現在システムでアクティブなルートを表示します。

```
# netstat -rn
```

[netstat\(1M\)](#) および [route\(1M\)](#) のマニュアルページを参照してください。

詳細は、[netstat\(1M\)](#) と [route\(1M\)](#) のマニュアルページを参照してください。

リアクティブモード使用時のデフォルトルートの作成および表示に関する情報は、[第5章「Oracle Solaris でのプロファイルベースのネットワーク構成の管理について」](#)を参照してください。

▼ ルーティングテーブルに静的ルートを追加する方法

1. 通常のユーザーアカウントを使用して、ルーティングテーブルの現在の状態を表示します。

```
% netstat -rn
```

出力は次のようになります。

```
Routing Table: IPv4
Destination      Gateway          Flags Ref    Use    Interface
-----
192.168.5.125    192.168.5.10    U      1      5879    net0
224.0.0.0        198.168.5.10    U      1        0      net0
default          192.168.5.10    UG     1     91908
127.0.0.1        127.0.0.1       UH     1     811302    lo0

Routing Table: IPv6
Destination/Mask  Gateway          Flags Ref    Use    If
-----
::1               ::1              UH     2        0     lo0
```

2. 管理者になります。
3. (オプション) ルーティングテーブル内の既存のエントリを消去します。

```
# route flush
```

4. 永続的なルートを追加します。

```
# route -p add -net network-address -gateway gateway-address
```

`-p` リブート後も保持されるルートを作成します。現在のセッションでのみ永続するルートを作成する場合は、`-p` オプションを使用しないでください。

`-net network-address` `network-address` で指定されたアドレスを持つネットワークへのルートであることを示します。

`-gateway gateway-address` 指定されたルートのゲートウェイシステムの IP アドレスが `gateway-address` であることを示します。

例 3-7 ルーティングテーブルに静的ルートを追加する

次の例は、ルーター (ルーター 2) に静的ルートを追加する方法を示します。静的ルートは AS のボーダールーター 10.0.5.150 が必要です。この特定の設定の図については、[図3-1「複数のIPv4 ルーターを備えた自律システム」](#)を参照してください。

次のようにして、ルーター 2 のルーティングテーブルを表示します。

```
# netstat -rn
Routing Table: IPv4
Destination          Gateway              Flags  Ref  Use  Interface
-----
default              172.20.1.10         UG     1    249  ce0
224.0.0.0            172.20.1.10         U      1     0    ce0
10.0.5.0             10.0.5.20          U      1    78    bge0
127.0.0.1            127.0.0.1          UH     1    57    lo0

Routing Table: IPv6
Destination/Mask     Gateway              Flags Ref  Use  If
-----
::1                  ::1                  UH    2    0    lo0
```

このルーティングテーブルは、ルーター 2 に既知のルートが 2 つあることを示しています。デフォルトのルートは、ルーター 2 の 172.20.1.10 インタフェースをゲートウェイとして使用します。2 番目のルート 10.0.5.0 は、ルーター 2 で実行中の in.routed デモンによって検出されました。このルートのゲートウェイはルーター 1 で、その IP アドレスは 10.0.5.20 です。

次のようにして、ボーダールーターとして機能するゲートウェイを持つネットワーク 10.0.5.0 に 2 番目のルートを追加します。

```
# route -p add -net 10.0.5.0/24 -gateway 10.0.5.150
add net 10.0.5.0: gateway 10.0.5.150
```

これで、IP アドレス 10.0.5.150/24 を持つボーダールーターへのルートが、ルーティングテーブルに追加されました。

```
# netstat -rn
Routing Table: IPv4
Destination          Gateway              Flags  Ref  Use  Interface
-----
default              172.20.1.10         UG     1    249  ce0
224.0.0.0            172.20.1.10         U      1     0    ce0
10.0.5.0             10.0.5.20          U      1    78    bge0
10.0.5.0             10.0.5.150         U      1   375    bge0
127.0.0.1            127.0.0.1          UH     1    57    lo0

Routing Table: IPv6
```

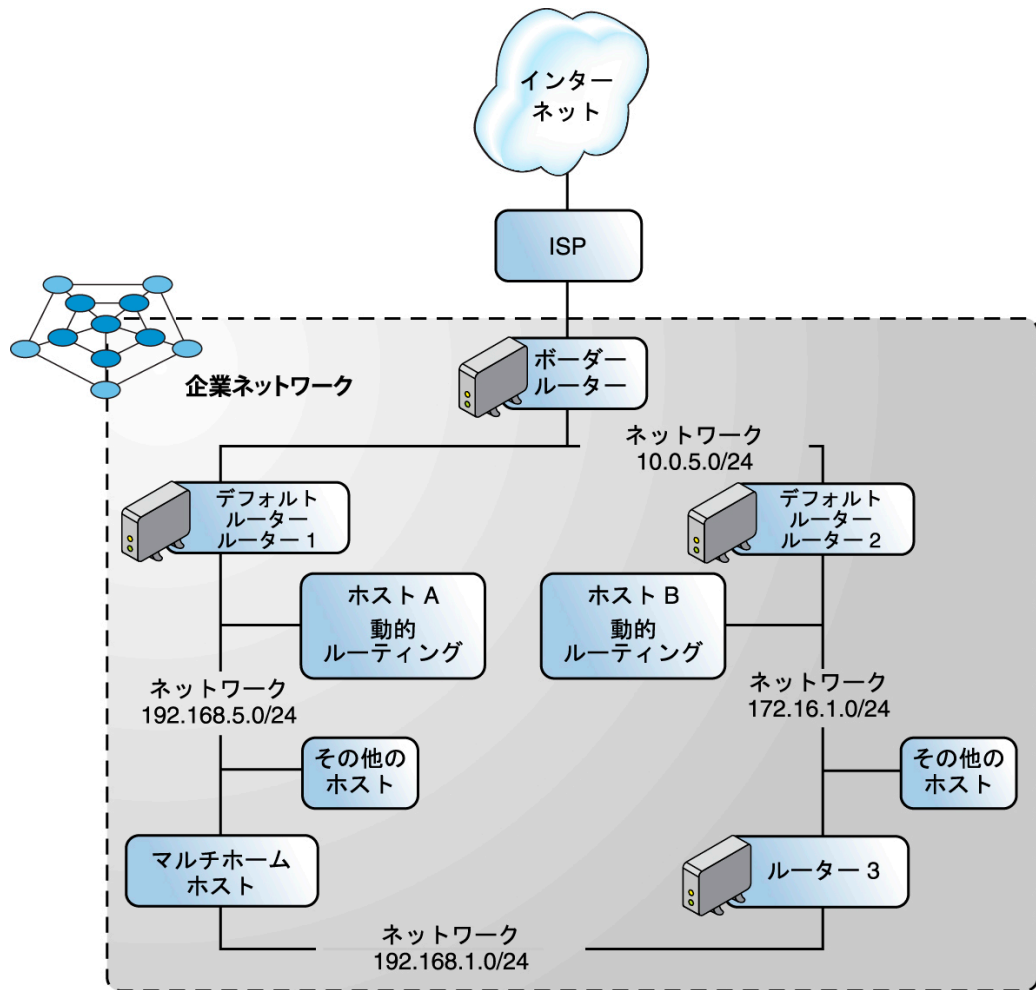
Destination/Mask	Gateway	Flags	Ref	Use	If
::1	::1	UH	2	0	lo0

単一インタフェースシステムのルーティングの有効化

単一インタフェースシステムは、静的ルーティングまたは動的ルーティングのいずれかで構成できます。静的ルーティングでは、ホストはデフォルトルーターのサービスを利用してルーティング情報を取得する必要があります。ルーティングプロトコルを使用する動的ルーティングの有効化は、システム上でルーティングを管理するもっとも簡単な方法です。

複数のルーターとネットワークを持つサイトでは、通常そのネットワークトポロジは単一のルーティングドメイン、つまり「自律システム」(AS: Autonomous System) として管理されます。このセクションの手順と例は、次の図をベースにしています。この図では、AS は 3 つのローカルネットワーク `10.0.5.0`、`172.20.1.0`、および `192.168.5.0` に分割されています。ネットワークはルーターとクライアントシステムで構成されており、これにはボーダールーター、デフォルトルーター、パケット転送ルーターが含まれます。クライアントシステムには、マルチホームシステムおよび単一インタフェースシステムが含まれます。これら各コンポーネントの詳細は、[『Oracle Solaris 11.2 でのネットワーク配備の計画』の「IPv4 自律システムのトポロジ」](#)を参照してください。

図 3-1 複数の IPv4 ルーターを備えた自律システム



▼ 単一インターフェースシステムで動的ルーティングを有効にする方法

次の手順は、システムの IP インタフェースがすでに構成されていることを想定しています。ネットワーク上でのルーターの計画に関する詳細は、『[Oracle Solaris 11.2 でのネットワーク配備の計画](#)』の「IPv4 自律システムのトポロジ」を参照してください。

1. 管理者になります。

2. システムが属するネットワークの IP アドレスを使用して、システムの IP インタフェースのいずれかを構成します。

手順については、51 ページの「IPv4 インタフェースを構成する方法」を参照してください。

3. 永続的に定義されたルートをすべてシステムから削除します。

この手順を実行するのは、静的に定義されたデフォルトルートが存在するため、システムのブート時にシステムが動的ルーティングを有効にできないためです。

- a. 次のようにして、永続的に定義されたデフォルトルートをすべて特定します。

```
# route -p show
```

- b. 永続的に定義された各ルートを削除します。例:

```
# route -p delete -net default -gateway 172.20.1.10
```

4. パケット転送が無効になっていることを確認します。

```
# routeadm -d ipv4-forwarding -u
```

5. システム上で IPv4 ルーティングを有効にします。

```
# routeadm -e ipv4-routing -u
```

例 3-8 単一インタフェースシステムで動的ルーティングを実行する

次の例は、図3-1「複数の IPv4 ルーターを備えた自律システム」に示すネットワーク 192.168.5.0 上にある単一インタフェースシステム hosta に動的ルーティングを構成する方法を示しています。システムはルーター 1 をデフォルトルーターとして使用します。この例は、システムの IP インタフェースがすでに構成されていることを想定しています。

まず、管理者の権利で hosta にログインします。次に、永続的に定義されたルートをすべてシステムから削除します。

```
# route -p show
persistent: route add default 172.20.1.10

# route -p delete default 172.20.1.10
delete net default: gateway 172.20.1.10
delete persistent net default: gateway 172.20.1.10

# routeadm
Configuration Current Current
Option Configuration System State
-----
```

```

IPv4 routing disabled disabled
IPv6 routing disabled disabled
IPv4 forwarding disabled disabled
IPv6 forwarding disabled disabled

Routing services "route:default ripng:default"

Routing daemons:

STATE FMRI
disabled svc:/network/routing/ripng:default
online svc:/network/routing/ndp:default
disabled svc:/network/routing/rdisc:default
disabled svc:/network/routing/legacy-routing:ipv4
disabled svc:/network/routing/legacy-routing:ipv6
disabled svc:/network/routing/route:default

# routeadm -d ipv4-forwarding -u
# routeadm -e ipv4-routing -u
# routeadm
Configuration Current Current
Option Configuration System State
-----
IPv4 routing enabled enabled
IPv6 routing disabled disabled
IPv4 forwarding disabled disabled
IPv6 forwarding disabled disabled

Routing services "route:default ripng:default"

Routing daemons:

STATE FMRI
disabled svc:/network/routing/ripng:default
online svc:/network/routing/ndp:default
disabled svc:/network/routing/rdisc:default
disabled svc:/network/routing/legacy-routing:ipv4
disabled svc:/network/routing/legacy-routing:ipv6
online svc:/network/routing/route:default

```

IPv6 ルーティングについて

IPv6 ルーティングは、CIDR における IPv4 ルーティングとほとんど同じです。唯一の違いは、IPv4 では 32 ビットアドレスを使用しますが、IPv6 では 128 ビットアドレスを使用することです。非常に簡単な拡張機能を使用することで、Open Shortest Path First (OSPF)、RIP、Inter-domain Routing Protocol (IDRP)、および Intermediate System to Intermediate System (IS-IS) などのすべての IPv4 ルーティングアルゴリズムを IPv6 のルーティングに使用できます。

IPv6 には、次に示す強力な新規ルーティング機能をサポートする、簡単なルーティング拡張機能も組み込まれました。

- プロバイダ選択 (ポリシー、パフォーマンス、コストなどを基準に)
- ホストの移動性 (現在の場所までのルート)
- アドレスの自動的な再指定 (新しいアドレスへのルート)

新しいルーティング機能を利用するには、IPv6 ルーティングオプションを使用する IPv6 アドレスのシーケンスを作成します。IPv6 の送信元は、ルーティングオプションを使用して、パケットが宛先に至るまでに経由する複数の中間ノード (またはトポロジカルグループ) をリストします。この中間ノードは、パケットの宛先の途中に通過します。この機能は、IPv4 での緩やかなソースと記録ルートオプションによく似ています。

アドレスシーケンスを一般的に使用する場合、(たいていは) ホストが受信したパケットのルートを逆戻りする必要があります。このパケットは、IPv6 認証ヘッダーを使用して正常に認証される必要があります。パケットを発信者に戻すには、アドレスシーケンスがパケット内に含まれている必要があります。IPv6 ホストの実装では、この方式により始点ルートの処理と逆引きをサポートしています。始点ルートの処理と逆引きにより、IPv6 の新機能 (プロバイダの選択や拡張アドレスなど) を実装するホストをプロバイダが使用することが可能になります。

マルチホームホストの構成

Oracle Solaris では、複数のインタフェースを持つシステムはマルチホームホストと見なされません。マルチホームホストのインタフェースは、異なる物理ネットワーク上または同じ物理ネットワーク上のさまざまなサブネットに接続します。マルチホームホストの詳しい作成手順については、[76 ページの「マルチホームホストの作成方法」](#)を参照してください。

複数のインタフェースが同じサブネットに接続しているシステムでは、最初にそれらのインタフェースを IPMP グループ内に構成する必要があります。そうしない場合、システムはマルチホームホストになることができません。IPMP の詳細は、『[Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理](#)』の第 2 章「[IPMP の管理について](#)」を参照してください。

マルチホームホストは IP パケットを転送しませんが、ルーティングプロトコルを実行するようにマルチホームホストを構成できます。一般に、次のような種類のシステムをマルチホームホストとして構成します。

- NFS サーバー、特に大規模なデータセンターとして機能する NFS サーバーを複数のネットワークに接続することによって、多数のユーザー間でファイルを共有できるようになります。この種のサーバーはルーティングテーブルを備えている必要はありません。

- データベースサーバーは、NFS サーバーと同様に、多数のユーザーにリソースを提供する複数のネットワークインタフェースを持つことができます。
- ファイアウォールゲートウェイは、企業のネットワークとインターネットなどの公共ネットワークとの間の接続を提供するシステムです。管理者は、セキュリティの手段としてファイアウォールを設定します。ファイアウォールとして構成されたホストは、ホストのインタフェースに接続されたネットワーク間でのパケット交換を行いません。ただしこの場合でも、承認ユーザーに対する ssh など、ホストは標準的な TCP/IP サービスを提供します。

注記 - マルチホームホストのいずれかのインタフェースでファイアウォールの種類が異なるときは、ホストのパケットの予期しない混乱を回避するようにしてください。この問題は、特にステートフルなファイアウォールで発生します。解決策の 1 つは、ステートレスなファイアウォールを構成することです。ファイアウォールの詳細は、『[Oracle Solaris 11.2 でのシステムおよび接続されたデバイスのセキュリティ保護](#)』の「[ファイアウォールシステム](#)」またはサードパーティー製のファイアウォールのドキュメントを参照してください。

▼ マルチホームホストの作成方法

1. 管理者になります。
2. インストールプロセスの一部として構成されなかった追加の各ネットワークインタフェースを構成します。

[51 ページの「IPv4 インタフェースを構成する方法」](#)を参照してください。

3. パケット転送が有効な場合、このサービスを無効にします。

```
# routeadm -p ipv4-forwarding
persistent=enabled default=disabled current=enabled
# routeadm -d ipv4-forwarding -u
# routeadm -p ipv4-forwarding
persistent=disabled default=disabled current=disabled
```

4. (オプション) マルチホームホストの動的ルーティングをオンに設定します。

```
# routeadm -e ipv4-routing -u
# routeadm -p ipv4-routing
persistent=enabled default=enabled current=enabled
```

例 3-9 マルチホームホストの構成

次の例は、『Oracle Solaris 11.2 でのネットワーク配備の計画』の「IPv4 自律システムのトポロジ」の図に示されているマルチホームホストを構成する方法を示します。この例で、システムのホスト名は `hostc` です。このホストには 2 つのインタフェースがあり、両方ともネットワーク `192.168.5.0` に接続されています。

まず、システムのインタフェースのステータスを表示します。

```
# dladm show-link
LINK      CLASS      MTU      STATE    BRIDGE    OVER
net0     phys      1500    up      --      --
net1     phys      1500    up      --      --

# ipadm show-addr
ADDROBJ   TYPE      STATE      ADDR
lo0/v4    static   ok         127.0.0.1/8
net0/v4    static   ok         192.168.5.82/24
```

`dladm show-link` コマンドの報告は、`hostc` に 2 つのデータリンクがあることを示しています。ただし、`net0` だけに IP アドレスが構成されています。`hostc` をマルチホームホストとして構成するには、`net1` に、同じ `192.168.5.0` ネットワーク内の IP アドレスを構成します。`net1` のベースとなる物理 NIC がネットワークに物理的に接続されていることを確認してください。

```
# ipadm create-ip net1
# ipadm create-addr static -a 192.168.5.85/24 net1
# ipadm show-addr
ADDROBJ   TYPE      STATE      ADDR
lo0/v4    static   ok         127.0.0.1/8
net0/v4    static   ok         192.168.5.82/24
net1/v4    static   ok         192.168.5.85/24
```

その後、次のように `net1` インタフェースを `/etc/hosts` ファイルに追加します。

```
# vi /etc/inet/hosts
127.0.0.1      localhost
192.168.5.82   hostc    #primary network interface for host3
192.168.5.85   hostc-2  #second interface
```

次に、パケット転送が `hostc` 上で実行中の場合、このサービスをオフにします。

```
# routeadm -p ipv4-forwarding
persistent=enabled default=disabled current=enabled

# routeadm

```

	Configuration Option	Current Configuration	Current System State
	IPv4 routing	enabled	enabled
	IPv6 routing	disabled	disabled
	IPv4 forwarding	disabled	disabled

```

IPv6 forwarding    disabled          disabled

Routing services   "route:default ripng:default"

Routing daemons:

STATE  FMRI
disabled  svc:/network/routing/ripng:default
online    svc:/network/routing/ndp:default
disabled  svc:/network/routing/rdisc:default
disabled  svc:/network/routing/legacy-routing:ipv4
disabled  svc:/network/routing/legacy-routing:ipv6
online    svc:/network/routing/route:default

```

routeadm コマンドの報告は、in.routed デモンによる動的ルーティングが現在有効になっていることを示しています。

マルチホームホストに対する対称ルーティングの実装

デフォルトでは、複数のインタフェースを備えたシステム (マルチホームホストとも呼ばれる) は、ルーティングテーブル内のトラフィックの宛先への最長一致ルートに基づいてネットワークトラフィックをルーティングします。宛先への長さが等しいルートが複数存在する場合、Oracle Solaris はこれらのルートにトラフィックを分散させるために等コストマルチパス (ECMP) アルゴリズムを適用します。

この方法でのトラフィックの分散が、常に最適であるというわけではありません。たとえば、ある IP パケットが、そのパケット内の IP 発信元アドレスと同じサブネット上に存在しないマルチホームホスト上のインタフェース経由で送信される可能性があります。さらに、送信パケットが特定の受信要求への応答 (ICMP エコー要求など) である場合は、要求と応答が同じインタフェースをたどらない可能性があります。このタイプのトラフィックルーティング構成を、*非対称ルーティング*と呼びます。インターネットサービスプロバイダ (ISP) が [RFC 3704 \(http://www.rfc-editor.org/rfc/bcp/bcp84.txt\)](http://www.rfc-editor.org/rfc/bcp/bcp84.txt) に記載されている *インGRESS フィルタリング* を実装している場合は、非対称ルーティング構成のために送信パケットが ISP によって破棄されることがあります。

RFC 3704 は、インターネット全体にわたるサービス拒否 (DoS) 攻撃の制限を目的としています。この目的に従うには、ネットワークを対称ルーティング用に構成する必要があります。IP の `hostmodel` プロパティを使用して、この要件を満たすことができます。このプロパティは、マルチホームホスト経由で受信または送信されている IP パケットの動作を制御します。

`hostmodel` プロパティは、3 つの値のいずれかに設定できます。

strong	RFC 1122 で定義されている強い終端システム (ES) モデルに対応します。この値によって、対称ルーティングが実装されます。
weak	RFC 1122 で定義されている弱い ES モデルに対応します。この値では、マルチホームホストは非対称ルーティングを使用します。
src-priority	優先ルートを使用してパケットルーティングを構成します。ルーティングテーブル内に複数の宛先ルートが存在する場合、優先ルートは、送信パケットの IP 発信元アドレスが構成されているインタフェースを使用するルートです。このようなルートが存在しない場合、送信パケットは、そのパケットの IP 宛先への一致する最長のルートを使用します。

たとえば、次のようにして、マルチホームホストに対して IP パケットの対称ルーティングを実装します。

```
# ipadm set-prop -p hostmodel=strong ipv4
# ipadm set-prop -p hostmodel=strong ipv6
# ipadm show-prop -p hostmodel ip
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
ipv6 hostmodel rw strong -- weak strong,
src-priority,
weak
ipv4 hostmodel rw strong -- weak strong,
src-priority,
weak
```

IP インタフェースプロパティとアドレスのカスタマイズ

IP インタフェースプロパティの管理に使用される 3 つの `ipadm` サブコマンドがあります。

- `show-ifprop -p property interface` – IP インタフェースのプロパティとその現在値を表示します。`-p property` オプションを使用しなかった場合は、IP インタフェースのすべてのプロパティが一覧表示されます。IP インタフェースを指定しなかった場合は、すべての IP インタフェースのすべてのプロパティが一覧表示されます。
- `set-ifprop -p property=value interface` – IP インタフェースのプロパティに値を割り当てます。
- `reset-ifprop -p property interface` – 特定のプロパティをそのデフォルト値にリセットします。

IP インタフェースにも、データリンクと同様に、特定のネットワーク環境に合わせてカスタマイズできるプロパティがあります。各インタフェースには、IPv4 プロトコル用と IPv6 プロトコル用の 2 つのプロパティセットが存在します。

MTU プロパティの設定

MTU プロパティを含む一部のプロパティは、データリンクと IP インタフェースの両方に共通です。そのため、データリンクに対してある MTU 値を構成し、そのリンク上に構成されているインタフェースに対して別の MTU 値を構成できます。さらに、その IP インタフェースをたどる IPv4 パケットと IPv6 パケットに適用される異なる MTU 値を構成できます。

IP インタフェースの MTU プロパティを設定する場合には、次の点に留意してください。

- IP インタフェースの MTU 設定の値を、データリンクの MTU 設定の値よりも大きくすることはできません。このような場合、`ipadm` コマンドはエラーメッセージを表示します。
- IP インタフェースの MTU 値がデータリンクの MTU 値とは異なる場合、IP パケットは IP インタフェースの MTU 値に制限されます。たとえば、データリンクの MTU 値が 9000 バイトで、IP インタフェースの MTU 値が 1500 バイトの場合、IP パケットは 1500 バイトに制限されます。ただし、ベースとなるレイヤー 2 プロトコルを使用しているほかのレイヤー 3 プロトコルは、最大 9000 バイトのパケットを送信できます。

データリンクの MTU 設定が IP インタフェースの MTU 設定に及ぼす影響などの情報を含む、データリンクプロパティのカスタマイズ手順については、[36 ページの「データリンクプロパティのカスタマイズ」](#)を参照してください。

パケット転送の有効化

ネットワーク上で、あるホストは、別のホストシステムに宛てられたデータパケットを受信できません。受信側のローカルシステムでパケット転送を有効にすることによって、そのシステムは、データパケットを宛先ホストに転送できます。このプロセスは *IP 転送* と呼ばれ、Oracle Solaris ではデフォルトで無効になっています。

パケット転送は、IP インタフェースと TCP/IP プロトコルのどちらにも設定できるプロパティによって管理されます。パケットの転送方法を選択できるようにする場合は、IP インタフェース上でパケット転送を有効にできます。たとえば、一部の NIC が外部ネットワークに接続されているのに対して、ほかの NIC はプライベートネットワークに接続されている場合のように、システムに複数の NIC が存在することがあります。そのため、すべてのインタフェースではなく、一部のインタフェース上でのみパケット転送を有効にします。

TCP/IP プロトコルのプロパティを設定することで、システム上でグローバルにパケット転送を有効化することもできます。詳細は、『[Oracle Solaris 11.2 での TCP/IP ネットワー](#)

ク、IPMP、および IP トンネルの管理』の「グローバルなパケット転送の有効化」を参照してください。

注記 - IP インタフェースとプロトコルの `forwarding` プロパティは、どちらも相互に排他的ではありません。インタフェースとプロトコルのプロパティを同時に設定できます。たとえば、プロトコル上でグローバルにパケット転送を有効にしてから、システム上の IP インタフェースごとにパケット転送をカスタマイズできます。したがって、そのシステムではパケット転送はグローバルに有効ですが、選択的でもあります。

たとえば、次のように IP インタフェースでパケット転送を有効にします。

```
# ipadm set-ifprop -p forwarding=on -m protocol-version interface
```

ここで、`protocol-version` は IPv4 または IPv6 のどちらかです。このコマンドは、IPv4 パケットと IPv6 パケットで別個に入力する必要があります。

次の例は、システム上で IPv4 パケット転送のみを有効にする方法を示します。

```
# ipadm show-ifprop -p forwarding net0
IFNAME  PROPERTY  PROTO  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
net0    forwarding  ipv4   rw    off      off         off      on,off
net0    forwarding  ipv6   rw    off      --         off      on,off

# ipadm set-ifprop -p forwarding=on -m ipv4 net0
# ipadm show-ifprop net0
IFNAME  PROPERTY  PROTO  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
...
net0    forwarding  ipv4   rw    on       on         off      on,off
...
```

IP アドレスプロパティのカスタマイズ

`ipadm` コマンドを使用すると、IP アドレス固有のプロパティを管理できます。

IP アドレスプロパティをカスタマイズして、次のネットワーク構成パラメータを管理できます。

- ネットマスクの長さ
- IP アドレスをアウトバウンドパケットの発信元アドレスとして使用できるかどうか
- アドレスが大域ゾーンまたは非大域ゾーンのどちらに属するか
- アドレスがプライベートアドレスであるかどうか

IP アドレスのプロパティを操作するときは、次の `ipadm` サブコマンドを使用します。

- `show-addrprop -p プロパティ addrobj` - 使用するオプションに応じて、アドレスプロパティを表示します。

すべての IP アドレスのプロパティを表示するには、プロパティおよびアドレスオブジェクトを指定しません。1 つのプロパティの値をすべての IP アドレスについて表示するには、そのプロパティのみを指定します。特定のアドレスオブジェクトのすべてのプロパティを表示するには、アドレスオブジェクトのみを指定します。

- `set-addrprop -p property=value addrobj` - アドレスプロパティに値を割り当てます。一度に設定できるアドレスのプロパティは 1 つだけです。
- `reset-addrprop -p property addrobj` - アドレスプロパティをデフォルト値に戻します。

注記 - 特定のインタフェースの IP アドレスを変更する場合は、`set-addressprop` サブコマンドを使用しないでください。代わりに、そのアドレスオブジェクトを削除し、新しい IP アドレスを使用して新しいオブジェクトを作成します。[84 ページの「IP インタフェース構成の削除または変更」](#)を参照してください。

たとえば、ある IP アドレスのネットマスクを変更するとします。この IP アドレスは、IP インタフェース `net3` 上に構成され、アドレスオブジェクト名 `net3/v4` で識別されます。次の例は、ネットマスクを改訂する方法を示しています。

```
# ipadm show-addr
ADDROBJ    TYPE      STATE    ADDR
lo0/?      static    ok       127.0.0.1/8
net3/v4     static    ok       192.168.84.3/24

# ipadm show-addrprop -p prefixlen net3/v4
ADDROBJ  PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
net3/v4  prefixlen rw     24       24          24        1-30,32

# ipadm set-addrprop -p prefixlen=8 net3/v4
# ipadm show-addrprop -p prefixlen net3/v4
ADDROBJ  PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
net3/v4  prefixlen rw     8        24          24        1-30,32
```

IP インタフェース構成の無効化、削除、および変更

このセクションには、次のトピックが含まれます。

- [83 ページの「IP インタフェース構成の削除」](#)
- [83 ページの「IP インタフェース構成の無効化」](#)
- [84 ページの「IP インタフェース構成の削除または変更」](#)

IP インタフェース構成の削除

`delete-ip` サブコマンドを使用して、構成済みの IP インタフェースを削除します。このコマンドは、特定のデータリンク構成タスクを実行している場合に特に重要です。たとえば、データリンク上に構成されている IP インタフェースが存在する場合に、そのデータリンクの名前変更が失敗します。データリンクの名前変更を試みる前に、`ipadm delete-ip` コマンドを使用して既存の IP 構成を削除する必要があります。次の例は、このタスクの実行で使用するコマンドを示します。

```
# ipadm delete-ip interface
# dladm rename-link old-name new-name
# ipadm create-ip interface
# ipadm create-address parameters
```

追加情報については、[35 ページの「データリンクの名前の変更」](#)も参照してください。データリンク名の変更後に IP インタフェースを再構成するには、[51 ページの「IPv4 インタフェースを構成する方法」](#)を参照してください。

IP インタフェース構成の無効化

デフォルトでは、`ipadm create-ip` コマンドを使用してインタフェースを作成するときに、IP インタフェースは `plumb` され、アクティブな構成の一部になります。インタフェース上で最初のアドレスを作成する際、インタフェースに `UP` のフラグが設定されます。

アクティブな構成を破棄せずに構成からインタフェースを削除するには、次のように `disable-if` サブコマンドを使用します。このサブコマンドは、カーネル内でインタフェースを `unplumb` します。

```
# ipadm disable-if -t interface
```

次のようにして、IP インタフェースを動作可能にし、そのフラグを `UP` として表示できます。

```
# ipadm enable-if -t interface
```

ヒント - IP インタフェースの現在のステータスを表示する場合は、[86 ページの「IP インタフェースに関する情報の取得」](#)を参照してください。

IP インタフェース構成の削除または変更

`ipadm delete-addr` コマンドは、特定のアドレス構成を IP インタフェースから削除します。このコマンドは、システムから単に IP アドレスを削除する場合、またはインタフェース上で構成されている IP アドレスの変更の一部として役立ちます。インタフェース上で構成されている IP アドレスを変更する場合は、最初に元のアドレス構成を削除してから、新しいアドレス構成を割り当てる必要があります。[84 ページの「既存の IP アドレスを変更する方法」](#)を参照してください。

インタフェースの IP アドレスの作成手順については、[51 ページの「IPv4 インタフェースを構成する方法」](#)を参照してください。

注記 - 1 つのインタフェースに複数の IP アドレスを構成できます。各アドレスは、アドレスオブジェクトによって識別されます。確実に正しいアドレスを削除するには、アドレスオブジェクトを知っている必要があります。`ipadm show-addr` コマンドを使用して、インタフェース上で構成されている IP アドレスを表示します。アドレスオブジェクトの説明については、[51 ページの「IPv4 インタフェースを構成する方法」](#)を参照してください。IP アドレスの表示の詳細は、[88 ページの「IP アドレスに関する情報の取得」](#)を参照してください。

▼ 既存の IP アドレスを変更する方法

次の手順では、システムの既存の IP アドレスを再構成する手順を説明します。

1. 管理者になります
2. 再構成する IP アドレスを表すアドレスオブジェクトを削除します。
3. 同じアドレスのオブジェクト名を使って、新しい IP アドレスを割り当てます。

```
# ipadm delete-addr addrobj
```

```
# ipadm create-addr -a IP-address addrobj
```

別のインタフェースをシステムに追加する場合は、[IPv4 インタフェースを構成する方法](#)を参照してください。

4. (オプション) 必要に応じて、次のようにシステムのホスト名を変更します。

```
# hostname new-hostname
```

5. (オプション) サブネットマスクが変更されている場合は、サブネットエントリを変更します。

6. (オプション) サブネットアドレスが変更されている場合は、デフォルトルーターの IP アドレスを変更します。
手順については、68 ページの「永続的 (静的) ルートの作成」を参照してください。
7. システムをリブートすると、変更内容が有効になります。

IP インタフェースとアドレスのモニタリング

IP インタフェースとそれらのプロパティに関する情報をモニターおよび取得するには、`ipadm` コマンドを使用します。このコマンドを単独で使用すると、システム上の IP インタフェースに関する一般情報が表示されます。ただし、次のコマンド構文を使用することにより、さまざまなサブコマンドを使用して表示する情報を制限することもできます。

`ipadm show-* other-arguments interface`

- インタフェース情報のみを取得するには、`show-if` サブコマンドを使用します。
- アドレス情報のみを取得するには、`show-addr` サブコマンドを使用します。
- インタフェースプロパティに関する情報を取得するには、`show-ifprop` サブコマンドを使用します。
- アドレスプロパティに関する情報を取得するには、`show-addrprop` サブコマンドを使用します。

`ipadm show-*` コマンドで表示されるすべてのフィールドの説明については、[ipadm\(1M\)](#) のマニュアルページを参照してください。

IP インタフェースに関する一般情報の取得

`ipadm` コマンドはシステムのインタフェースの全体像を提供します。このコマンドをサブコマンドを付けずに使用すると、システムのすべての IP インタフェースに関するデフォルトの情報が提供されます。例:

```
# ipadm
NAME          CLASS/TYPE STATE   UNDER ADDR
lo0           loopback  ok      --    --
lo0/v4        static    ok      --    127.0.0.1/8
lo0/v6        static    ok      --    ::1/128
net0          ip        ok      --    --
net0/v4       static    ok      --    10.132.146.233/23
```

```

net0/v4      dhcp      ok        --      10.132.146.234/23
ipmp0       ipmp      degraded --      --
ipmp0/v6    static    ok        --      2001:db8:1:2::4c08/128
net1        ip        failed    ipmp0   --
net1/v6     addrconf ok        --      fe80::124:4fff:fe58:1831/10
net2        ip        ok        ipmp0   --
net2/v6     addrconf ok        --      fe80::214:4fff:fe58:1832/10
iptun0      ip        ok        --      --
iptun0/v4   static    ok        --      172.16.111.5->172.16.223.75
iptun0/v6   static    ok        --      fe80::10:5->fe80::223:75
iptun0/v6a  static    ok        --      2001:db8:1a0:7::10:5->2001:db8:7a82:64::223:75

```

前の出力には、次の情報が表示されています。

- IP インタフェース。
- 各インタフェースのクラス。
- 各インタフェースの状態。
- インタフェースのステータス (「スタンドアロン」の IP インタフェースであるか、別のインタフェース構成タイプのベースとなるインタフェースであるか)。この例では、net1 と net2 は、UNDER 列に示されているように ipmp0 のベースとなるインタフェースです。
- インタフェースに関連付けられたアドレスオブジェクト。アドレスオブジェクトは特定の IP アドレスを識別します。これらのアドレスオブジェクトは、「NAME」見出しの下に一覧表示され、インタフェース名と区別するためにインデントされます。
- 「CLASS/TYPE」見出しの下にインデントされる IP アドレスのタイプで、static や dhcp などがあります。
- 「ADDRESS」列の下に一覧表示される実際のアドレス。

IP インタフェースに関する情報の取得

IP インタフェースに関する情報を取得するには、ipadm show-if *interface* コマンドを使用します。インタフェースを指定しない場合は、この情報にシステム上のすべてのインタフェースが含まれます。

コマンド出力の各フィールドは、次の情報を示します。

IFNAME	情報が表示されているインタフェースを示します。
CLASS	インタフェースのクラスを示します。次の 4 つのいずれかです。 <ul style="list-style-type: none"> ■ ip は IP インタフェースを示します。 ■ ipmp は IPMP インタフェースを示します。

- vni は仮想インタフェースを示します。
- loopback は、自動的に作成されるループバックインタフェースを示します。ループバックインタフェースを除き、残りの 3 つのインタフェースクラスは手動で作成できます。

STATE	<p>インタフェースのステータスを示します。ok、offline、failed、down、disabled のいずれかです。</p> <p>failed のステータスは IPMP グループに適用され、ダウンしているためにトラフィックをホストできないデータリンクまたは IP インタフェースを示す場合があります。IP インタフェースが IPMP グループに属している場合、IPMP インタフェースは、グループ内のほかのアクティブな IP インタフェースを使用してトラフィックを引き続き送受信できます。</p> <p>down のステータスは、管理者によってオフラインに切り替えられた IP インタフェースを示します。</p> <p>disable のステータスは、ipadm disable-if コマンドを使用して unplumb されている IP インタフェースを示します。</p>
ACTIVE	<p>インタフェースがトラフィックをホストするために使用されているかどうかを示し、yes または no のどちらかに設定されます。</p>
OVER	<p>インタフェースの IPMP クラスにのみ適用され、IPMP インタフェースまたはグループを構成するベースとなるインタフェースを示します。</p>

次に、このコマンドで表示される情報の例を示します。

```
# ipadm show-if
IFNAME      CLASS      STATE      ACTIVE      OVER
lo0         loopback   ok         yes         --
net0        ip         ok         yes         --
net1        ip         ok         yes         --
tun0        ip         ok         yes         --
```

IP インタフェースのプロパティに関する情報の取得

ipadm show-ifprop *interface* コマンドを使用して、IP インタフェースのプロパティに関する情報を取得します。プロパティまたはインタフェースを指定しなかった場合は、システム上のすべての IP インタフェースのすべてのプロパティに関する情報が表示されます。

コマンド出力の各フィールドは、次の内容を示します。

IFNAME 情報が表示されている IP インタフェースを示します。

PROPERTY	インタフェースのプロパティを示します。インタフェースには、複数のプロパティが含まれる場合があります。
PROTO	プロパティが適用されるプロトコル (IPv4 または IPv6 のどちらか) を示します。
PERM	指定されたプロパティの許可されるアクセス権を示します。読み取り専用、書き込みのみ、またはその両方のいずれかです。
CURRENT	アクティブな構成内のプロパティの現在値を示します。
PERSISTENT	システムがリブートされたときに再適用されるプロパティの値を示します。
DEFAULT	指定されたプロパティのデフォルト値を示します。
POSSIBLE	指定されたプロパティに割り当てることができる値の一覧を示します。数値の場合は、受け入れ可能な値の範囲が表示されます。

注記 - いずれかのフィールド値が不明である場合 (たとえば、情報を要求されているプロパティがインタフェースでサポートされていない場合)、その値は疑問符 (?) として表示されません。

次の例は、`show-ifprop` サブコマンドで表示される情報のタイプを示します。

```
# ipadm show-ifprop -p mtu net1
IFNAME PROPERTY PROTO PERM CURRENT PERSISTENT DEFAULT POSSIBLE
net1    mtu      ipv4  rw   1500    --      1500    68-1500
net1    mtu      ipv6  rw   1500    --      1500    1280-1500
```

IP アドレスに関する情報の取得

IP アドレスに関する情報を取得するには、`ipadm show-addr interface` コマンドを使用します。インタフェースを指定しない場合は、システム上のすべての IP アドレスに関する情報が表示されます。

コマンド出力の各フィールドは、次の内容を示します。

ADDROBJ	IP アドレスが表示されるアドレスオブジェクトを指定します。
TYPE	IP アドレスが <code>static</code> 、 <code>dhcp</code> 、 <code>addrconf</code> のいずれであるかを示します。 <code>addrconf</code> の値は、アドレスがステートレスまたはステートフルアドレス構成を使用して取得されたことを示します。

STATE	アクティブな構成内のアドレスオブジェクトのステータスを示します。これらの値の完全な一覧については、 ipadm(1M) のマニュアルページを参照してください。
ADDR	<p>インタフェース上に構成されている IP アドレスを指定します。このアドレスは IPv4 または IPv6 のどちらかです。トンネルインタフェースでは、ローカルアドレスとリモートアドレスの両方が表示されます。</p> <p>トンネルに関する詳細は、『Oracle Solaris 11.2 での TCP/IP ネットワーク、IPMP、および IP トンネルの管理』の第 5 章「IP トンネルの管理」を参照してください。</p>

次に、`show-addr` サブコマンドで提供される情報の例を示します。

```
# ipadm show-addr
ADDROBJ      TYPE      STATE      ADDR
lo0/v4       static    ok         127.0.0.1/8
net0/v4       static    ok         192.168.84.3/24
tun0/v4       static    ok         172.16.134.1-->172.16.134.2
```

このコマンドにインタフェースを指定し、そのインタフェースに複数のアドレスがある場合は、次のような情報が表示されます。

```
# ipadm show-addr net0
ADDROBJ      TYPE      STATE      ADDR
net0/v4       static    ok         192.168.84.3/24
net0/v4a      static    ok         10.0.1.1/24
net0/v4bc     static    ok         172.16.10.1
```

`interface/?` として表示されるアドレスオブジェクトは、そのアドレスが `libipadm` API を使用しなかったアプリケーションによってインタフェース上に構成されたことを示しています。このようなアプリケーションは、アドレスオブジェクト名に `interface/user-defined-string` という形式の使用を要求する `ipadm` コマンドの制御下にはありません。IP アドレスの割り当ての例については、[51 ページの「IPv4 インタフェースを構成する方法」](#)を参照してください。

IP アドレスのプロパティに関する情報の取得

IP アドレスプロパティに関する情報を取得するには、`ipadm show-addrprop addrobj` コマンドを使用します。すべてのプロパティを一覧表示するには、`addrobj` オプションを省略します。1 つのプロパティをすべての IP アドレスについて一覧表示するには、そのプロパティのみを指定します。特定のアドレスのすべてのプロパティを表示するには、`addrobj` オプションのみを指定します。

コマンド出力の各フィールドは、次の内容を示します。

ADDROBJ	プロパティが表示されているアドレスオブジェクトを示します。
PROPERTY	アドレスオブジェクトのプロパティを示します。アドレスオブジェクトには、複数のプロパティが含まれる場合があります。
PERM	指定されたプロパティの許可されるアクセス権を示します。読み取り専用、書き込みのみ、またはその両方のいずれかです。
CURRENT	現在の構成内のプロパティの実際の値を示します。
PERSISTENT	システムがリブートされたときに再適用されるプロパティの値を示します。
DEFAULT	指定されたプロパティのデフォルト値を示します。
POSSIBLE	指定されたプロパティに割り当てることができる値の一覧を示します。数値の場合は、受け入れ可能な値の範囲が表示されます。

次の例は、show-addrprop サブコマンドで表示される情報のタイプを示します。

```
# ipadm show-addrprop net1/v4
ADDROBJ  PROPERTY  PERM  CURRENT      PERSISTENT  DEFAULT      POSSIBLE
net1/v4  broadcast r-    192.168.84.255  --          192.168.84.255  --
net1/v4  deprecated rw     off          --          off           on,off
net1/v4  prefixlen rw     24          24          24           1-30,32
net1/v4  private  rw     off          --          off           on,off
net1/v4  transmit rw     on          --          on           on,off
net1/v4  zone     rw     global       --          global        --
```

◆◆◆ 第 4 章

Oracle Solaris クライアントでのネームサービスとディレクトリサービスの管理

この章では、Oracle Solaris ホストクライアントシステムのネームサービスを構成する方法を説明します。ネームサービスとディレクトリサービス、およびサーバー側の管理の完全な概要については、『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』を参照してください。

ネームサービスとディレクトリサービスの構成のトラブルシューティングについては、『Oracle Solaris 11.2 でのネットワーク管理のトラブルシューティング』の第 3 章「ネームサービスの問題のトラブルシューティング」を参照してください。

この章の内容は、次のとおりです。

- 91 ページの「ネームサービス構成の新機能」
- 92 ページの「ネームサービスとディレクトリサービスの構成の概要」
- 95 ページの「システムをローカルファイルモード用に構成する」
- 97 ページの「DNS クライアントの構成」
- 100 ページの「NIS クライアントの構成」
- 102 ページの「LDAP クライアントの構成」

注記 - 特に明記されていない場合、この章で説明するタスクは、IPv4 と IPv6 の両方のネットワークに適用されます。

ネームサービス構成の新機能

次の機能が新しく導入または変更されています。

- ネームサービスおよびシステム構成の **SMF** への移行 - このリリースでは、ネームサービスはサービス管理機能 (SMF) を介して管理されます。ネームサービスを構成するために `/etc/`

nsswitch.conf や /etc/resolv.conf などの特定ファイルを変更するというこれまでの動作は、もはや有効ではありません。この Oracle Solaris リリースでは、レガシー構成ファイルは、以前の Oracle Solaris リリースとの互換性維持のためにのみ保持されています。これらのファイルの内容は、特定のネームサービスと関連する SMF サービスにより生成されます。

ネットワーク構成が存在しない場合、ネームサービスは nis files ではなく files only 動作にデフォルト設定されます。svc:/system/name-service/cache SMF サービスは常時有効にするべきです。SMF コマンドを使用してこれらのサービスの構成を変更する場合、サービスを有効にするかリフレッシュして、あるいはその両方を実行して、変更を有効にする必要があります。svccfg(1M) および svcadm(1M) のマニュアルページを参照してください。

- **resolv.conf のエラーチェック機能** – SMF へのネームサービスの移行前に、resolv.conf ファイル構成内のエラーがサイレントに処理され、警告を生成せずに検出されない状態になりました。その結果、resolv.conf ファイルは構成されたとおりに動作しませんでした。Oracle Solaris 11 では、エラー状態の報告が適切に実行されるように、SMF テンプレートを使用した基本エラーチェックが導入されています。resolv.conf(4) のマニュアルページを参照してください。
- **Domain Name System (DNS) サーバーの設定** – DNS サーバーの設定プロセスが変更されました。詳しい手順については、『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の「DNS の管理 (タスク)」を参照してください。

ネームサービスとディレクトリサービスの構成の概要

ネームサービスは、ホスト名とアドレス、ユーザー名、パスワード、アクセス権などの格納されている情報の検索を実行します。これらの情報はユーザーに使用可能になるため、ユーザーは各自のホストにログインし、リソースにアクセスして、アクセス許可を取得することができます。ネームサービス情報は、ファイル、マップ、または各種の形式のデータベースファイルに格納できます。これらの情報リポジトリは、システムにローカルに格納することも、中央のネットワークベースのリポジトリまたはデータベースに格納することもできます。中央のネームサービスが存在しない場合、各ホストは、これらの情報の独自のコピーを保持する必要があります。すべてのデータを 1 か所で管理すれば、管理がより簡単になります。ネームサービスは、どのようなコンピュータネットワークにも欠かせないものです。

次のネームサービスとディレクトリサービスがサポートされています。

- **ドメインネームサービス (DNS)**

DNS は、TCP/IP ネットワーク上に実装された階層的な分散型データベースです。これは主に、インターネットホスト名に対する IP アドレス、および IP アドレスに対するホスト名を検索するために使用されます。これらのデータはネットワーク全体にわたって分散しており、右から左に読み取られる、ピリオドで区切られた名前を使用して検索されます。DNS はまた、メール交換のルーティング情報、場所のデータ、使用可能なサービスなどの、その他のインターネット関連のホスト情報を格納するためにも使用されます。このサービスの階層的な性質により、ローカルドメインのローカルでの管理が可能になるだけでなく、インターネット、イントラネット、またはその両方に接続されたほかのドメインに国際的に対処できるようになります。詳しい手順については、『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の「DNS ネームサービスの説明」を参照してください。

DNS プロトコルに対する 2 つの拡張機能が `svc:network/dns/multicast` サービスにより管理されています。マルチキャスト DNS (mDNS) では、従来の DNS サーバーがインストールされていなかった小規模なネットワーク内に DNS が実装されます。また、DNS サービス検出 (DNS-SD) によって、マルチキャスト DNS が単純なサービス検出 (ネットワークブラウジング) を提供するように拡張されます。『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の「マルチキャスト DNS およびサービス検出の説明」を参照してください。

■ ネットワーク情報システム (NIS)

NIS (このガイドでは「ニス」と発音) は、DNS とは別個に開発されました。NIS は、多様なネットワーク情報を集中管理することによりネットワーク管理機能を高めることに焦点を当てています。NIS には、ネットワーク、マシンの名前とアドレス、ユーザー、およびネットワークサービスに関する情報も格納されます。このようなネットワーク情報の集まりを「NIS の名前空間」と呼びます。詳細は、『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の「NIS ネームサービスの説明」を参照してください。

■ LDAP (Lightweight Directory Access Protocol)

LDAP は、ディレクトリサーバーにアクセスして分散型ネームサービスやその他のディレクトリサービスを使用するために使用される、セキュアなネットワークプロトコルです。この標準ベースのプロトコルは、階層的なデータベース構造をサポートしています。同じプロトコルを使用して、UNIX とマルチプラットフォームの両方の環境でネームサービスを提供できます。Oracle Solaris は、Oracle Directory Server Enterprise Edition (従来の Sun Java System Directory Server) およびほかの LDAP Directory Server を使用する場合、LDAP をサポートします。詳細は、『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の「LDAP ネームサービスの説明」を参照してください。

Oracle Solaris でのネームサービスサポート (サーバー側およびクライアント側) の完全な概要については、『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の第 1 章「ネームサービスとディレクトリサービスについて」および『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: LDAP』を参照してください。

name-service/switch SMF サービスについて

name-service/switch SMF サービスは、構成可能な選択サービスで、これを使って各タイプのネットワーク情報に使用するネーム情報サービスまたはソースを指定できます。

ネームサービススイッチは、次のインタフェースのいずれかを呼び出すクライアントアプリケーションにより使用されます。

- gethostbyname
- getpwuid
- getpwnam
- getaddrinfo

name-service/switch SMF サービスは、各ネットワークデータベースで使用する 1 つまたは複数のネームサービスを定義します。この情報は、以前は /etc/nsswitch.conf ファイルに格納されていました。このファイルは引き続き存在しますが、ファイルに含まれる構成を変更するには、SMF サービス内の適切なプロパティを変更する必要があります。

これらのプロパティは、次のようにして表示できます。

```
$ svccfg -s name-service/switch listprop config
config                application
config/default        astring              files
config/value_authorization astring              solaris.smf.value.name-service.switch
config/password        astring              "files ldap"
config/group           astring              "files ldap"
config/host            astring              "files dns"
config/automount       astring              "files ldap"
```

config/default プロパティには、検索対象のデフォルトソースを 1 つまたは複数指定します。特定のデータベースに独自のプロパティセットが存在しない場合は、1 つまたは複数のデフォルトソースが使用されます。前の例では、password、group、host、および automount を除くすべてのデータベースが、ローカルファイルをソースとして使用します。デフォルト以外の 1 つまたは複数のソースが必要な場合、特定のデータベース用のプロパティが作成されます。この例では、password、groups、および automount が、最初にローカルファイル内で、次に LDAP 内で検索されます。ホスト検索が、最初にローカルファイル内で、次に DNS 内で実行されます。

システムで有効にされているネームサービスを変更する場合、name-service/switch SMF サービスの適切なプロパティを更新して、正しいネームサービスを使用する必要があります。たとえば、name-service/switch を前の例と同じように構成した場合には、LDAP を無効にして、代わりに NIS を有効にします。

- この場合、name-service/switch サービスの次のプロパティを設定して、ファイルおよび NIS を使用する必要があります。
- config/password
- config/group
- config/automount

次のコマンドを入力して、これらのプロパティを正しく設定します。

```
# svccfg -s name-service/switch setprop config/password = astring: "files nis"
# svccfg -s name-service/switch setprop config/group = astring: "files nis"
# svccfg -s name-service/switch setprop config/automountconfig/password = astring: "files
nis"
# svccfg -s name-service/switch:default refresh
```

詳細は、『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の第 2 章「ネームサービススイッチについて」および『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の「ネームサービススイッチの構成」を参照してください。

システムをローカルファイルモード用に構成する

ローカルファイルモードで動作するときは、システムはローカルディレクトリにあるファイルからすべての TCP/IP 構成情報を取得します。ネットワーククライアントモードでは、構成情報はリモートネットワーク構成サーバーによって、ネットワーク内のすべてのシステムに提供されます。

一般的に、ネットワーク内の次のサーバーはローカルファイルモードで動作します。

- ネットワーク構成サーバー
- NFS サーバー
- NIS、LDAP、または DNS のサービスを提供するネームサーバー
- メールサーバー
- ルーター

クライアントは、どのネットワーク上でもネットワーククライアントモードまたはローカルファイルモードのいずれかで動作可能であるため、これらのモードを組み合わせ、さまざまなシステムを構成できます。

▼ システムをローカルファイルモード用に構成する方法

1. 管理者になります。
2. システムの IP インタフェースに、割り当て済み IP アドレスを構成します。
[IPv4 インタフェースを構成する方法](#)を参照してください。

3. ホスト名が正しく設定されたことを確認します。

```
# hostname
```

詳細は、[hostname\(1\)](#) のマニュアルページを参照してください。

4. `/etc/inet/hosts` 内のエントリが最新であることを確認します。

Oracle Solaris は、プライマリネットワークインタフェース、ループバックアドレス、およびインストール時に構成された追加インタフェース (該当する場合) に対するエントリを作成します。エントリが最新ではない場合、インストール後にシステムに追加されたネットワークインタフェースの IP アドレスとそれに対応する名前を追加します。

5. システムの完全修飾ドメインを `nis/domain` SMF サービスのプロパティとして指定します。
たとえば、次のように `nis/domain` SMF サービスの `domainname` プロパティの値として `deserts.worldwide.com` を指定します。

```
# domainname domainname
```

このステップで、変更が永続的なものになります。

6. ルーティング情報を追加します。

注記 - DHCP サービスを使用している場合は、このステップをスキップできます。

手順については、[66 ページの「ルーティングの構成」](#)を参照してください。

7. ネットマスク情報を追加します (該当する場合)。

注記 - DHCP サービスを使用している場合は、このステップをスキップできます。

- a. ネットワーク番号とネットマスクを `/etc/inet/netmasks` ファイルに入力します。

エントリを作成するには、`network-number netmask` の形式を使用します。たとえば、Class C ネットワーク番号 192.168.83 を指定するには、次の情報を入力します。

```
192.168.83.0    255.255.255.0
```

CIDR アドレスの場合は、ネットワークの接頭辞をそれと同等の 10 進ドット表記に変換します。たとえば、192.168.3.0/22 という CIDR ネットワーク接頭辞を表現するには、次の情報を入力します。

```
192.168.3.0    255.255.252.0
```

- b. ローカルのファイルのみが検索されるように `name-service/switch` プロパティーのネットマスク検索ソースを変更してから、インスタンスをリフレッシュします。

```
# svccfg -s name-service/switch setprop config/netmask = astring: "files"
# svccfg -s name-service/switch:default refresh
```

8. システムをリブートします。

DNS クライアントの構成

DNS には、答えを提供するサービスと、そのサービスに照会するクライアントという 2 つの部分があります。Oracle Solaris では、デフォルト DNS サービスは、Internet Systems Consortium (ISC) の Berkeley Internet Name Domain (BIND) と、それに関連付けられた `named` サーバーデーモンにより提供されます。DNS クライアントは、ユーティリティーとライブラリの集まりで構成されます。

タスク関連の追加情報については、『[Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS](#)』の「DNS の管理 (タスク)」を参照してください。

▼ DNS クライアントを有効にする方法

1. 管理者になります。

2. 検索するドメインと、DNS ネームサーバーの IP アドレスを一覧表示してから、SMF リポジトリを更新します。例:

```
# svccfg -s network/dns/client
svc:/network/dns/client> setprop config/search = astring: ("example.com"
"sales.example.com")
svc:/network/dns/client> setprop config/nameserver = net_address: (192.168.1.10
192.168.1.11)
svc:/network/dns/client> select network/dns/client:default
svc:/network/dns/client:default> refresh
svc:/network/dns/client:default> quit
```

変更を有効にするには、必ずサービスをリフレッシュしてください。

3. DNS を使用するようにネームサービススイッチ情報を更新します。

最初のコマンドによって、SMF リポジトリ内の DNS 構成情報が更新されます。

```
# svccfg -s system/name-service/switch
svc:/system/name-service/switch> setprop config/host = astring: "files dns"
svc:/system/name-service/switch> select system/name-service/switch:default
svc:/system/name-service/switch:default> refresh
svc:/system/name-service/switch:default> quit
```

4. DNS クライアントを実行するために必要なサービスを起動します。

```
# svcadm enable network/dns/client
# svcadm enable system/name-service/switch
```

5. 次のコマンドの 1 つまたは両方を使用して、DNS クライアントが有効になっていることを確認します。

```
# dig knownserver.example.com
# getent hosts knownserver.example.com
```

dig コマンドは、単独で使用された場合には、DNS クライアントが有効であることをチェックします。getent hosts コマンドは、`/etc/nsswitch.conf` ファイルの DNS クライアントの使用を検証します。

例 4-1 クライアントの複数の DNS オプションを同時に設定する

次の例は、複数の `/etc/resolv.conf` オプションを設定する方法を示しています。

```
# svccg
svc:> select /network/dns/client
svc:/network/dns/client> setprop config/options = "ndots:2 retrans:3 retry:1"
svc:/network/dns/client> listprop config/options
config/options astring      ndots:2 retrans:3 retry:1
```

```
svc:/network/dns/client> exit
# svcadm refresh dns/client
# grep options /etc/resolv.conf
options ndots:2 retrans:3 retry:1
```

マルチキャスト DNS の有効化

マルチキャスト DNS (mDNS) および DNS サービスディレクトリが動作するには、mDNS に参加するすべてのシステムで mDNS が配備されている必要があります。mDNS サービスは、システム上で提供されているサービスの可用性を通知するために使用されます。

mDNS を有効にする前に、ソフトウェアパッケージがシステムにインストールされていることを確認します。必要に応じ、次のようにパッケージをインストールします。

```
# pkg install pkg:/service/network/dns/mdns
```

mDNS の有効化プロセスの一部として、最初にネームサービススイッチ情報を更新します。ローカルホストを解決できるようにするため、次のように name-service/switch SMF サービスの config/host プロパティを変更して、ソースとして mdns を含めます。

```
# /usr/sbin/svccfg -s svc:/system/name-service/switch
svc:/system/name-service/switch> setprop config/host = astring: "files dns mdns"
svc:/system/name-service/switch> select system/name-service/switch:default
svc:/system/name-service/switch:default> refresh
svc:/system/name-service/switch> quit
```

次のようにして mDNS SMF サービスを有効にします。

```
# svcadm enable svc:/network/dns/multicast:default
```

この方法で mDNS を有効にすると、アップグレードおよびリブートしても変更は持続します。詳細は、[svcadm\(1M\)](#) のマニュアルページを参照してください。

DNS のためのリソースの通知

dns-sd コマンドを使用して、ping や traceroute コマンドの使用方法和似た方法でサービスを参照および検出できます。dns-sd コマンドは主に、対話的に使用されます。その主な理由として、そのコマンド行引数や出力形式が時間とともに変化する場合があります、それがシェルスクリプトからのコマンドの起動を予測不可能で、かつ危険なものにしていることが挙げられます。さらに、DNS サービス検出 (DNS-SD) は非同期の性質を持っているため、スクリプト指向のプログラミングには適していません。

たとえば、『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の「DNS のためのリソースの通知」を参照してください。[dns-sd\(1M\)](#) のマニュアルページも参照してください。

NIS クライアントの構成

NIS は、ネットワークオブジェクトおよびリソースの識別と検索に使用される分散型ネームサービスです。これは、ネットワーク全体の情報に関する一様なストレージと検索方法を、トランスポートプロトコルやメディアに依存しない形式で提供します。NIS は DNS から独立して開発されたため、その焦点は少し異なっています。DNS は数値 IP アドレスの代わりにマシン名を使うことによって、通信を簡略化することに焦点を当てているのに対して、NIS の場合は、多様なネットワーク情報を集中管理することによりネットワーク管理機能を高めることに焦点を絞っています。

NIS を実行することにより、管理データベース (マップ) をさまざまなサーバー間 (マスターとスレーブ) で分散できます。さらに、これらの管理データベースを一元管理により自動的かつ確実な方法で更新できるため、どのクライアントもネットワーク全体を通して一貫した方法で同じネームサービス情報を共有できます。詳しい概要については、『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の第 5 章「ネットワーク情報サービスについて」を参照してください。

システムで NIS クライアントを提供する SMF サービスは 2 つあります。svcadm コマンドを使用して、これらのサービスを有効化、無効化、再起動、およびリフレッシュします。次のように、システム上の NIS サービスの状態を表示します。

```
# svcs \*nis\*
STATE          STIME    FMRI
online         Oct_09   svc:/network/nis/domain:default
online         Oct_09   svc:/network/nis/client:default
```

タスク関連の追加情報については、『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の第 6 章「ネットワーク情報サービスの設定および構成」を参照してください。

▼ ブロードキャストモードで NIS クライアントを構成する方法

ブロードキャストモードは、NIS クライアントを確立するもっとも簡単な方法です。nis/client SMF サービスを起動すると、このサービスが ypbind コマンドを実行し、これにより NIS サーバーのローカルサブネットが検索されます。サブネットが見つかり、ypbind コマンドがそこに

バインドします。この検索をブロードキャストと呼びます。クライアントのローカルサブネット上に NIS サーバーが存在しない場合は、ypbind コマンドがバインドに失敗し、クライアントマシンは NIS サービスから名前空間データを取得できません。101 ページの「特定の NIS サーバーを使用して NIS クライアントを構成する方法」を参照してください。

1. 管理者になります。
2. NIS ドメイン名を設定します。

```
# domainname example.com
```

3. 必要に応じて、ネームサービススイッチに変更を加えます。
『Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: DNS と NIS』の「ネームサービススイッチの構成」を参照してください。
4. NIS クライアントサービスを起動します。

```
# svcadm enable network/nis/domain  
# svcadm enable network/nis/client
```

▼ 特定の NIS サーバーを使用して NIS クライアントを構成する方法

始める前に 次の手順では、手順 3 で指定されるホスト名 (サーバー) を DNS で解決できることが必要です。DNS を使用しておらず、IP アドレスの代わりにホスト名を入力する場合は、クライアント上の /etc/hosts ファイルに各 NIS サーバーの適切なエントリを必ず追加してください。詳細は、[ypinit\(1M\)](#) のマニュアルページを参照してください。

1. 管理者になります。
2. NIS ドメインを設定します。

```
# domainname example.com  
# svcadm enable network/nis/domain
```

3. クライアント構成スクリプトを実行します。

```
# ypinit -c
```

クライアントがネームサービス情報を取得する元の NIS サーバーを指定するよう求められます。マスターサーバーと、必要な数のスレーブサーバーをリストできます。指定するサーバーはドメイン

内のどこにあってもかまいません。最初に、マシンに (ネットワーク的に) もっとも近いサーバーをリストし、次にネットワークのより遠い部分に存在するサーバーをリストすることをお勧めします。

4. NIS クライアントを有効にします。

```
# svcadm enable network/nis/client
```

▼ NIS クライアントサービスを無効にする方法

1. 管理者になります。

2. 次のようにして、NIS クライアントサービスを停止します。

```
# svcadm disable network/nis/domain  
# svcadm disable network/nis/client
```

LDAP クライアントの構成

Oracle Solaris クライアントで LDAP をネームサービスとして使用するには、次の要件が満たされている必要があります。

- クライアントのドメイン名が LDAP サーバーによって処理されている必要があります。
- ネームサービススイッチが、必要なサービスの LDAP を指し示している必要があります。
- クライアントに、その動作を定義する特定のパラメータがすべて構成されている必要があります。
- `ldap_cachemgr` がクライアント上で実行されている必要があります。
- クライアントが構成されているサーバーが少なくとも 1 つ起動され、実行されている必要があります。

`ldapclient` コマンドは、サーバーの起動を除き、上記のタスクをすべて実行するので、LDAP クライアントを設定するための鍵となります。

ネットワーク構成に固定モードを使用する場合、クライアントシステム上で LDAP を設定するもっとも簡単な方法は、`DefaultFixed` プロファイルを有効にして、永続ネットワーク構成を実行することです。次に、`ldapclient` コマンドを使用し、プロファイルまたは手動設定を利用して、LDAP の設定を完了できます。詳細は、[ldapclient\(1M\)](#) のマニュアルページを参照してください。

LDAP の詳しい概要については、『[Oracle Solaris 11.2 ディレクトリサービスとネームサービスでの作業: LDAP](#)』を参照してください。

ネームサービス構成のインポート

`nscfg` コマンドは、`name-service` スイッチコンポーネントの旧バージョンファイルの構成を SMF リポジトリに転送します。Oracle Solaris 11 へのアップグレード時に、システムのネームサービス構成が自動的に SMF に移行します。ただし必要に応じ、`nscfg` コマンドを使用して、手動でこの構成を SMF に移行できます。

次のコマンドは、旧バージョンのファイルをインポートし、構成を変換して SMF にプッシュします。

```
# /usr/sbin/nscfg import -f FMRI
```

`nscfg` コマンドの使用は、既存の `resolv.conf` ファイルの情報を使用して、DNS 構成を生成するもっとも簡単な方法です。次の例では、`nscfg` コマンドは `/etc/resolv.conf` ファイルの情報を読み取って変換したあと、`svc:/network/dns/client` SMF サービス内に情報を格納します。

```
# cp resolv.conf /etc/resolv.conf
# /usr/sbin/nscfg import -f dns/client
# svcadm enable dns/client
```

システムのネームサービスを変更した場合、ネームサービススイッチの情報を適宜変更する必要もあり、次の例に示すように、場合によっては古い情報をネームサービスキャッシュからフラッシュする必要もあります。

```
# cp /etc/nsswitch.dns /etc/nsswitch.conf
# /usr/sbin/nscfg import -f name-service/switch
# svcadm refresh name-service/switch
# svcadm refresh name-service/cache
```

詳細は、[nscfg\(1M\)](#) のマニュアルページを参照してください。

SMF ネームサービス構成のリセット

SMF ネームサービス構成の構成プロパティを、次のように `files only` モードにリセットできます。

```
# /usr/sbin/nscfg unconfig FMRI
# svcadm refresh name-service/switch
```

たとえば、次のようにして、[103 ページの「ネームサービス構成のインポート」](#)で SMF 構成に加えた変更を元に戻します。

```
# svcadm disable dns/client
# /usr/sbin/nscfg unconfig dns/client
# /usr/sbin/nscfg unconfig name-service/switch
# svcadm refresh name-service/switch
# svcadm refresh name-service/cache
```


◆◆◆ 第 5 章

Oracle Solaris でのプロファイルベースのネットワーク構成の管理について

この章では、システムでリアクティブモードが使用されている場合に主に使用される、プロファイルベースのネットワーク構成の概要を示します。リアクティブモードは、Oracle Solaris システム上のネットワーク構成を管理するための、さまざまなタイプのプロファイルをサポートします。このモードがもっとも多用されるのは、ノートブック PC およびネットワーク状態が頻繁に変化する状況です。リアクティブモードでは、ネットワークデーモン (nwmnd) がシステムの状態をモニターして、状態が変化するとネットワーク構成を動的に調整します。

企業環境でもっとも多用されるネットワーク管理モデルである固定モードの使用については、[第2章「Oracle Solaris でのデータリンク構成の管理」](#)および[第3章「Oracle Solaris での IP インタフェースとアドレスの構成および管理」](#)を参照してください。

ネットワークプロファイルの構成および管理に関する詳細な手順は、[第6章「Oracle Solaris でのプロファイルベースのネットワーク構成の管理」](#)を参照してください。

この章の内容は、次のとおりです。

- [106 ページの「リアクティブモードについて」](#)
- [107 ページの「プロファイルベースのネットワーク構成について」](#)
- [113 ページの「プロファイルベースのネットワーク構成の使用ガイドライン」](#)
- [115 ページの「プロファイルベースのネットワーク構成を使用するためのセキュリティー要件」](#)
- [116 ページの「プロファイルベースのネットワーク構成がほかの Oracle Solaris 機能と連携して動作する方法」](#)

リアクティブモードについて

リアクティブモードでは、システムは、手動の再構成を必要とせず、自動的にネットワーク状態の変化に適応して、システムの現在のネットワーク構成を調整します。たとえば、有線ネットワークインタフェースが取り外された場合、新しい無線ネットワークが使用可能になった場合、または物理的な場所が変化した場合、それに応じてシステムがネットワーク構成を適応させます。

Oracle Solaris のリアクティブネットワーク構成ポリシーでは、機動性に重点が置かれ、さまざまなネットワークイベントやユーザーのリクエストに応じてシステムのネットワーク構成を動的に変更できます。このタイプのネットワーク構成がもっとも有効に機能するのは、ノートブック PC およびネットワーク状態が頻繁に変化する状況です。リアクティブモードの使用時には、システムの基本 Ethernet および WiFi 構成が自動的に実行されます。システムは、起動時に有線または無線ネットワークに自動的に接続し、現在アクティブなネットワーク接続のステータスに関する通知をデスクトップ上に表示します。特定のプロファイルが有効になる条件を決定するプロパティを含むリアクティブプロファイルを構成できます。これらのプロパティは、必要に応じてネットワーク管理デーモン `nwamd` を使用して、プロファイルの構成をシステムに動的に適用させることができます。

通常、リアクティブネットワーク構成の変更は、次のイベントおよびアクティビティで呼び出されます。

- Ethernet ケーブルを接続または切断する
- 無線ローカルエリアネットワーク (WLAN) カードの接続または切断
- 有線インタフェースまたは無線インタフェースが使用可能なときにシステムをブートする
- 有線インタフェースまたは無線インタフェースが使用可能な (サポートされている) ときに中断状態から再開する
- DHCP リースを取得または解放する

リアクティブモードでは、ネットワーク構成の管理に、プロファイルにネットワーク構成変更を加える `netcfg` コマンドと、プロファイルの情報の表示およびプロファイルの有効化と無効化を実行する `netadm` コマンドの 2 つを使用します。詳細な説明は、[netcfg\(1M\)](#) および [netadm\(1M\)](#) のマニュアルページを参照してください。タスク関連の情報については、[第6章「Oracle Solaris でのプロファイルベースのネットワーク構成の管理」](#)を参照してください。

一方、固定モードはリアクティブモードの反対です。固定モードを使用している場合、ネットワークデーモンは、システム上で特定のネットワーク構成をインスタンス化しますが、その構成をさまざまなネットワーク状態に合わせて自動的に調整することはしません。固定モードの詳細は、[17 ページの「ネットワーク構成モードについて」](#)を参照してください。

サポートされているすべてのネットワーク管理コマンドおよびそれらをいつ使用するかに関する詳細は、[18 ページの「Oracle Solaris ネットワーク管理コマンド」](#)を参照してください。

プロファイルベースのネットワーク構成について

Oracle Solaris は、事前決定済みのシステム定義プロファイル、および特定ネットワークのニーズに合わせてプロパティやアクティブ化条件を指定してさまざまなタイプのユーザー定義リアクティブプロファイルを作成する機能を提供します。ユーザー定義プロファイルを使用すると、システムのデータリンクや IP アドレスの構成を簡略化したり、ネームサービス、IP フィルタ、IP セキュリティー (IPsec) 構成などのより複雑なシステム規模のネットワーク構成を定義したりできます。

次のプロファイルタイプがサポートされています。

- **ネットワーク構成プロファイル (NCP)** – NCP は、ネットワークデータリンクや IP インタフェースの構成の指定に使用される主なプロファイルタイプです。NCP の構成に使用するプロパティ値は、特定の NCP がシステムでアクティブである場合のネットワーク構成方法を指定します。NCP はリアクティブまたは固定にできます。複数のリアクティブ NCP を構成できますが、Oracle Solaris でサポートされる固定 NCP は DefaultFixed という名前のもので 1 つだけです。
- **ネットワーク構成単位 (NCU)** – NCP を定義する個別の構成情報 (プロパティ) は、NCU 内部で指定されます。NCU は、物理リンクまたはインタフェースを表し、そのリンクまたはインタフェースの構成を指定するプロパティを含めることができます。
- **場所プロファイル** – 場所プロファイル (場所とも呼ばれる) は、ネームサービス、ドメイン、IP フィルタ構成、および IPsec 構成などのシステム規模のネットワーク構成を指定します。
- **外部ネットワーク修飾子 (ENM)** – ENM は、VPN アプリケーションなどの、システムのプライマリネットワーク構成外部にあるネットワーク構成の作成を担当するアプリケーションを管理するプロファイルです。
- **既知の WLAN** – 既知の WLAN は、システムにより検出される無線ネットワークに関する情報を格納するプロファイルです。

プロファイルタイプの説明

次に、Oracle Solaris リリースでサポートされている各プロファイルタイプの詳細な説明を示します。

NCP の説明

NCP は、データリンクおよび IP インタフェースやアドレスなどの、システム固有のネットワーク構成を定義します。各 NCP の一部であるさまざまな NCU (ネットワーク構成単位) は、起動されるインタフェース (複数可)、インタフェースが起動される条件、インタフェースの IP アドレスを取得する方法などの、さまざまなネットワークリンクおよびインタフェースの構成方法を指定します。

Automatic NCP は、現在システムに存在するすべてのネットワークリンクおよびインタフェースを表します。ネットワークデバイスが追加または削除されると、Automatic NCP の内容が変更されます。Automatic NCP は、DHCP アドレス自動構成を利用するプロファイルへのアクセスを提供するため、システムの IP アドレスの取得が可能になります。この NCP には、無線リンクよりも有線リンクが優先されるリンク選択ポリシーも実装されています。代替の IP 構成ポリシー、または代替のリンク選択ポリシーの指定が必要な場合は、別の NCP をシステムに作成する必要があります。Automatic NCP は削除できません。この NCP をコピーし、コピーに変更を加えることができます。例6-7「[Automatic NCP をクローニングして NCP を作成する](#)」を参照してください。

NCU の説明

NCU には、NCP を定義するプロパティ値が含まれます。NCU は、システムに存在する個々の物理リンクおよびインタフェースを表します。ユーザー定義の NCP を構成するプロセスには、各リンクおよびインタフェースを構成する方法および条件を指定する NCU の作成が伴います。

NCP には次の 2 種類があります。

- リンク NCU - 物理デバイス (開放型システム間相互接続 (OSI) モデルの第 2 層エンティティ) を表します。
- インタフェース NCU - IP インタフェース (第 3 層エンティティ) を表します

リンク NCU は、次のデータリンク層クラスを表します。

- アグリゲーション
- ブリッジ
- Etherstub
- EoIB (Ethernet over IB)、
- 物理リンク (Ethernet または WiFi)
- トンネル

- 仮想拡張ローカルエリアネットワーク (VXLAN)
- 仮想ローカルエリアネットワーク (VLAN)
- 仮想ネットワークインタフェースカード (VNIC)

インタフェース NCU は、次の IP 層クラスを表します。

- IP インタフェース
- IPMP インタフェース
- VNI インタフェース

さまざまなオブジェクトタイプに設定可能なプロパティについては、[netcfg\(1M\)](#) のマニュアルページを参照してください。

場所プロファイルの説明

場所プロファイル (単に場所とも呼ばれる) には、ネームサービスやファイアウォール設定などのネットワーク構成情報が含まれ、その場所がアクティブな場合に、これらがまとめて適用されてシステム規模のネットワーク構成を指定します。場所は必ずしも物理的な場所に対応しているわけではないため、さまざまなネットワーク要件を満たすように複数の場所プロファイルを設定できます。たとえば、会社のイントラネットに接続するときに、1 つの場所を使用できます。オフィスに配置されている無線アクセスポイントを使用してパブリックインターネットに接続するときに、もう 1 つの場所を使用できます。

デフォルトでは、3 つの場所プロファイルがシステムで事前定義されています。

■ DefaultFixed

DefaultFixed の場所は、DefaultFixed NCP がアクティブな場合にはいつでも有効です。DefaultFixed の場所は、netcfg コマンドを使用して直接変更することはできません。この場所が有効な場合 (DefaultFixed NCP の有効化の一部として)、関連するサービス管理機能 (SMF) プロパティが更新されて、場所の設定が反映されます。システムがシャットダウンしているか、別の場所が有効な場合、関連する SMF プロパティが DefaultFixed の場所の構成の一部として保存されます。

■ Automatic

場所 Automatic は、使用可能なネットワークが存在するが、それより優先されるほかの場所が存在しない場合にアクティブになります。netcfg コマンドを使用して、場所 Automatic を変更できます。

注記 - 場所 Automatic と Automatic NCP を混同しないでください。場所 Automatic は、システムの初期ネットワーク構成が実行されたあとでシステム全体のネットワークプロパティを定義します。Automatic NCP は、システムのリンクおよびインタフェースのネットワーク構成を指定します。

■ NoNet

場所 NoNet には、非常に具体的なアクティブ化条件があります。この場所は、IP アドレスが割り当てられているローカルインタフェースがない場合に、システムによってスタンドアロンシステムに適用されます。netcfg コマンドを使用して、場所 NoNet を変更できます。

ユーザー定義の場所はユーザーが指定した値で構成されるのに対して、システム定義の場所には事前設定された値があるという点を除いて、ユーザー定義の場所はシステム定義の場所と同じです。

ENM の説明

ENM では、アプリケーションまたはスクリプトが、NCP および場所プロフィールで指定された構成の外部のネットワーク構成をいつ行うかを指定できます。ENM は、有効または無効になったときにネットワーク構成を直接変更するサービスやアプリケーションとして定義することもできます。ENM を有効または無効にする条件を指定できます。ENM を手動で有効または無効にすることもできます。常にシステムで各プロフィールタイプの 1 つのみをアクティブにできる NCP または場所プロフィールとは異なり、同時に複数の ENM がシステムでアクティブになる可能性があります。常にシステムでアクティブな ENM は、同時にシステムで有効になる NCP または場所プロフィールに必ずしも依存しません。

ENM を作成できる外部アプリケーションおよびサービスはいくつか存在しますが、典型的な例として VPN アプリケーションが挙げられます。システムに VPN をインストールして構成したあとに、指定した条件で VPN が自動的にアクティブおよび非アクティブになるように、ENM を作成できます。

注記 - リアクティブネットワーク構成モードは、システムのネットワーク構成を直接変更する機能を持つ外部アプリケーションを自動的に検出できません。VPN アプリケーション、または任意の外部アプリケーションやサービスのアクティブ化または非アクティブ化を管理するには、まずアプリケーションをインストールしてから、コマンド行インタフェース (CLI) とネットワーク管理 GUI のどちらかを使用して、そのアプリケーション用の ENM を作成する必要があります。

ENM で実行されるネットワーク構成に関する永続的な情報が格納または追跡される方法は、NCP または場所プロファイルに関する情報が格納される方法とまったく同じではありません。ただし、システムには、外部で開始されたネットワーク構成を記述し、ENM によってシステムに行われた構成の変更に基づいて、どの場所をアクティブにする必要があるかどうかを再評価したあとに、その場所をアクティブにする機能があります。たとえば、特定の IP アドレスが使用中のときに、条件付きでアクティブ化される場所に切り替えることがあります。いつでも `svc:/network/physical:default` サービスが再起動されると、アクティブな NCP で指定されたネットワーク構成が回復します。同様に ENM が再起動されると、プロセス中にネットワーク構成が破棄され、再作成される可能性があります。

既知の WLAN の説明

既知の WLAN は、システムに認識されている無線ネットワークを管理するために使用されるプロファイルです。次に、これらの既知の無線ネットワークのグローバルなリストは、システムにより保守されます。この情報は、使用可能な無線ネットワークへの接続を試みる順序を決定する際に使用されます。既知の WLAN リストに存在する無線ネットワークが使用可能な場合、システムは自動的にそのネットワークに接続します。既知の無線ネットワークを 2 つ以上使用可能な場合、システムは優先度がもっとも高い (番号がもっとも小さい) 無線ネットワークへの接続を試みます。接続する新しい無線ネットワークは自動的に既知の WLAN リストのいちばん上に追加され、現在優先度がもっとも高い無線ネットワークになります。

デフォルト動作では、以前に接続された WLAN よりも最近接続された WLAN の方が優先されます。既知の WLAN が同じ優先度を共有することは決してありません。既存の WLAN と同じ優先度値を持つ新規 WLAN がリストに追加されると、既存のエントリが低い優先度値にシフトします。その結果、リスト内のその他すべての WLAN の優先度値が動的に低い優先度値にシフトします。

既知の WLAN に、1 つの鍵名を関連付けることもできます。鍵名があれば、`dladm create-secobj` コマンドを使用して独自の鍵を作成できます。その後、WLAN keyname プロパティにセキュアなオブジェクト名を追加すると、この鍵を WLAN に関連付けることができます。詳細は、[dladm\(1M\)](#) のマニュアルページを参照してください。

コマンドからの WLAN の管理に関する詳細は、[159 ページの「リアクティブモードでの既知の WLAN の管理」](#)を参照してください。

システム定義およびユーザー定義のプロファイル

Automatic NCP は、システムに存在する物理リンクごとに、1 つのリンク NCU と 1 つのインタフェース NCU で構成されるシステム定義のプロファイルです。この NCP の NCU アクティブ化ポリシーでは、無線リンクよりも有線リンクの接続が優先され、有効な各リンクで IPv4 と IPv6 の両方が plumb されます。IPv4 アドレスを取得するには、DHCP が使用されます。IPv6 アドレスを取得するには、ステートレス自動構成および DHCP が使用されません。Automatic NCP は、新しいリンクがシステムに挿入またはシステムから削除されると動的に変更されます。挿入または削除されたリンクに対応するすべての NCU も、同時に追加または削除されます。プロファイルは、`nwamd` デモンによって自動的に更新されます。

アクティブな場合、Automatic NCP は次の基本ポリシーを実装します。

- DHCP を使用して、使用可能な (接続されている) すべての Ethernet インタフェースを構成します。
- Ethernet インタフェースが接続されていない場合や、IP アドレスを取得できない場合は、1 つの無線インタフェースを有効にすると、既知の WLAN リストから最適な WLAN が選択され自動的に接続されます。[111 ページの「既知の WLAN の説明」](#)を参照してください。
- 1 つ以上の IP4 アドレスが取得されるまで、場所 NoNet はアクティブのままにします。[109 ページの「場所プロファイルの説明」](#)を参照してください。この場所には、IP アドレスの取得に関連するデータ (DHCP および IPv6 autoconf メッセージ) のみを渡すという厳格な IP 振り分け規則があります。場所 NoNet のすべてのプロパティは、アクティブ化条件を除いて変更可能です。
- 1 つ以上の IP アドレスがシステムのインタフェースのいずれかに割り当てられたら、場所 Automatic をアクティブ化します。この場所には、IP フィルタや IPsec の規則はありません。場所には、DHCP サーバーから取得されたドメインネームシステム (DNS) 構成データが適用されます。場所 NoNet と同様に、場所 Automatic のすべてのプロパティは、アクティブ化条件を除いて変更可能です。

オプションでユーザー定義 NCP を構成できます。指定された NCP から NCU を明示的に追加および削除する必要があります。現在システムに存在するリンクに関連しない NCU を作成することもできます。さらに、ユーザー定義の NCP のポリシーを決定することもできます。たとえば、特定の時間にシステムで複数のリンクおよびインタフェースが有効になることを許可したり、NCU と静的 IP アドレス間にさまざまな依存関係を指定したりできます。

プロファイルベースのネットワーク構成の使用ガイドライン

プロファイルベースのネットワーク構成は、次のガイドラインに従います。

- システムで一度にアクティブにできるのは、1 つのネットワーク構成プロファイル (NCP) および 1 つの場所プロファイルだけです。システム上のほかの既存の NCP はすべて、非運用状態になります。
- アクティブな NCP は、リアクティブまたは固定 (DefaultFixed) のいずれかになります。リアクティブ NCP では、システムのネットワーク環境の変更に適応できるように、システムがネットワーク構成をモニターします。DefaultFixed NCP (システム唯一の固定プロファイル) では、ネットワーク構成はインスタンス化されますが、モニターされません。
- NCP のさまざまなプロパティの値が、そのプロファイルによるシステムのネットワーク構成管理方法を制御するポリシーを構成します。
- NCP のプロパティに対する変更は、新しいプロパティ値として即座に実装され、そのプロファイルがアクティブになる場合には常にネットワーク構成を管理するプロファイルのポリシーの一部となります。
- システムでリアクティブモードが使用されている場合、ネットワーク構成を管理するアクティブな NCP は Automatic NCP または作成するユーザー定義のリアクティブ NCP のいずれかになります。リアクティブ NCP がアクティブな場合、netcfg および netadm コマンドを使用して、ネットワーク構成を管理します。

システムで固定モードが使用されている場合は、ネットワーク構成を管理するアクティブな NCP は常に DefaultFixed になります。この NCP がアクティブな場合、dladm および ipadm コマンドを使用してネットワーク構成を管理します。詳細は、[第2章「Oracle Solaris でのデータリンク構成の管理」](#)および[第3章「Oracle Solaris での IP インタフェースとアドレスの構成および管理」](#)を参照してください。

プロファイルのアクティブ化ポリシー

プロファイルのアクティブ化モードは、手動、自動、または条件付きのいずれかです。リアクティブプロファイルがアクティブな場合、ネットワーク環境の変化によりシステムがネットワーク構成を再評価し、現在の条件に基づいてリアクティブ NCP および場所のアクティブ化に関する最適な推測を行います。ネットワーク状態の変化には、Ethernet ケーブルの接続または切断、DHCP リースの取得または解放、および新しい無線ネットワークの検出が含まれます。常に、アクティブな NCP と場所がシステムに 1 つ存在する必要があります。

リアクティブネットワーク構成モードでは、リアクティブ NCP のアクティブ化ポリシーを指定できます。このポリシーは、NCU を有効にするタイミングを決定します。個々の場所プロフィールにも、アクティブ化の条件を定義したプロパティが含まれています。

NCU、場所プロフィール、および ENM には、どれにも `activation-mode` プロパティがあります。許容される値は、プロフィールタイプごとに異なります。さらに、`activation-mode` プロパティを検証する方法、および各プロフィールが有効になる条件も、プロフィールタイプごとに異なります。

注記 - NCU の `activation-mode` プロパティは、`manual` と `prioritized` のどちらかに設定できます。場所プロフィールの `activation-mode` プロパティは、`manual`、`conditional-any`、`conditional-all`、または `system` に設定できます。

NCP アクティブ化ポリシーは、NCU ごとに指定可能なプロパティと条件を使用することで適用されます。指定されるポリシーの例として、「無線接続よりも有線接続を優先する」や「一度に 1 つのインタフェースをアクティブにする」などがあります。NCP を有効にする方法とタイミングは、NCU タイプごとに設定されたプロパティで定義されます。アクティブ化条件の詳細は、[netcfg\(1M\)](#) のマニュアルページを参照してください。

注記 - インタフェース NCU はベースとなるリンク NCU に関連付けられています。各インタフェース NCU は、関連付けられたリンク NCU を有効にするとアクティブになります。ベースとなるリンク NCU との依存関係は削除できません。関連付けられたリンク NCU を有効化せずにインタフェース NCU を有効化する場合は、そのインタフェースのベースとなるリンク NCU が有効になるまで、インタフェース NCU は実際にアクティブになりません。

プロフィールのアクティブ化条件

ユーザー定義の NCU、場所プロフィール、および ENM にはすべて、`activation-mode` プロパティが存在します。`netcfg` コマンドを使用してプロフィールを作成または変更すると、`activation-mode` プロファイルが設定されます。NCP に `activation-mode` プロパティはありません。すべての NCP は手動で有効化されます。

次の表に、さまざまなプロフィールタイプの `activation-mode` プロパティの取り得る値を示します。

表 5-1 `activation-mode` プロパティの値

プロフィールタイプ	<code>activation-mode</code> の値
NCU	<code>manual</code> または <code>prioritized</code>

プロファイルタイプ	activation-mode の値
場所	manual、conditional-any、conditional-all、または system
ENM	manual、conditional-any、または conditional-all

プロファイルの有効化および無効化の詳細は、[119 ページの「プロファイルを有効および無効にする」](#)を参照してください。

プロファイルベースのネットワーク構成を使用するためのセキュリティ要件

netcfgd デーモンは、すべてのネットワーク構成情報が格納されるリポジトリを制御します。netcfg コマンド、ネットワーク管理 GUI、および nwamd デーモンはそれぞれ、netcfgd デーモンにリクエストを送信してリポジトリにアクセスします。

現在のネットワーク構成実装では、特定のタスクを実行する際に次の承認が使用されます。

- solaris.network.autoconf.read – netcfgd デーモンで検証されるネットワークプロファイルデータの読み取りを有効にします。
- solaris.network.autoconf.write – netcfgd デーモンで検証されるネットワークプロファイルデータの書き込みを有効にします。
- solaris.network.autoconf.select – nwamd デーモンで検証される新しい構成データの適用を有効にします。
- solaris.network.autconf.wlan – 既知の WLAN 構成データの書き込みを有効にします。

これらの承認は、auth_attr データベースに登録されています。[auth_attr\(4\)](#) のマニュアルページを参照してください。

solaris.network.autoconf.read 承認は、すべてのユーザーにデフォルトで割り当てられる Basic Solaris User 権利プロファイルに含まれています。したがって、この承認を持つユーザーは、ネットワークの現在の状態とすべてのネットワークプロファイルの内容を表示できます。

追加の権利プロファイルとして、Network Autoconf User と Network Autoconf Admin の 2 つがあります。Network Autoconf User プロファイルには、read、select、および

wlan 承認があります。Network Autoconf Admin プロファイルには、write 承認が追加されます。Network Autoconf User プロファイルは、Console User プロファイルに割り当てられます。デフォルトでは、コンソールにログインしているユーザーならだれでも、プロフィールの表示、有効化、および無効化が可能です。Console User プロファイルには solaris.network.autoconf.write 承認が割り当てられていないため、この承認を持つユーザーは NCP、NCU、場所、または ENM を作成または変更できません。ただし、Console User プロファイルは WLAN を表示、作成、および変更することはできます。

netcfg および netadm コマンドは、Basic Solaris User 権利プロフィールを持つユーザーがネットワークプロフィールを表示するために使用できます。このプロフィールは、デフォルトですべてのユーザーに割り当てられます。

netadm コマンドは、Network Autoconf User または Console User プロファイルを持つユーザーがプロフィールを有効にするために使用することもできます。Console User プロファイルは、/dev/console からシステムにログインしているユーザーに自動的に割り当てられます。

netcfg コマンドを使用してネットワークプロフィールを変更するには、solaris.network.autoconf.write 承認または Network Autoconf Admin プロファイルが必要です。

たとえば、次のようにして、Console User 権利プロフィールに関連付けられた特権を特定します。

```
$ profiles -p "Console User" info
name=Console User
desc=Manage System as the Console User
auths=solaris.system.shutdown,solaris.device.cdrw,solaris.devinde.mount.removable,
solaris.smf.manage.vbiosd,solaris.smf.value.vbiosd
profiles=Suspend To RAM,Suspend To Disk,Brightness,CPU
Power Management,Network Autoconf User,Desktop Removable Media User
help=RtConsUser.html
```

プロフィールベースのネットワーク構成がほかの Oracle Solaris 機能と連携して動作する方法

リアクティブネットワーク構成モードは、次のようにして、ほかの Oracle Solaris ネットワークテクノロジーと連携して動作します。

- 仮想マシン: Oracle VM Server for SPARC (以前の論理ドメイン) および Oracle VM VirtualBox

リアクティブプロファイルは、Oracle Solaris ホストとゲストの両方と連携して動作します。ただし、リアクティブネットワーク構成モードは、システムのほかの仮想マシンに干渉することなく、特定の仮想マシンに属するインタフェースのみを管理します。

■ Oracle Solaris ゾーンとスタックインスタンス

リアクティブプロファイルは、大域ゾーンまたは排他的なスタック (非大域) ゾーンで動作します。ただし、共有スタックゾーンのネットワーク構成は常に大域ゾーンで管理されるため、共有スタックゾーン用のリアクティブプロファイルは構成できません。

■ 動的再構成 (DR)

システムのネットワーク構成は、動的再構成 (DR) 機能とホットプラグ機能を保持するシステムに対して、これらの機能をサポートします。これらの機能を使用して、現在アクティブな NCP (リアクティブ NCP または DefaultFixed NCP) に関係なく、デバイスの追加や削除を実行できます。ただし、システムの動作は、現在アクティブな NCP のタイプによって異なります。

Automatic NCP または別のリアクティブ NCP がアクティブで、デバイスが接続されている場合、NCP は新たに追加されたデバイスの IP 構成を自動的に作成します。リアクティブプロファイルが現在アクティブな状態で、デバイスがシステムから取り外された場合、そのデバイスの IP インタフェースは構成解除されます。DefaultFixed NCP がシステムでアクティブである場合、デバイスの追加後に IP インタフェースを明示的に構成する必要があります。また、デバイスを取り外す前に、IP 構成を明示的に削除する必要があります。

デバイスの動的構成の詳細は、『[Oracle Solaris 11.2 でのデバイスの管理](#)』を参照してください。固定プロファイル使用時の動的再構成の実行に関する詳細は、[動的再構成を使用してネットワークインタフェースカードを交換する方法](#)を参照してください。

◆◆◆ 第 6 章

Oracle Solaris でのプロフィールベースのネットワーク構成の管理

この章では、さまざまなタイプのプロファイルを使用して、ネットワーク構成を管理する方法について説明します。プロフィールベースのネットワーク構成およびリアクティブモードの概要については、[第5章「Oracle Solaris でのプロフィールベースのネットワーク構成の管理について」](#)を参照してください。

固定モード使用時のデータリンクおよび IP インタフェースの構成については、[第2章「Oracle Solaris でのデータリンク構成の管理」](#)および[第3章「Oracle Solaris での IP インタフェースとアドレスの構成および管理」](#)を参照してください。

この章の内容は、次のとおりです。

- [119 ページの「プロフィールを有効および無効にする」](#)
- [121 ページの「プロフィールの構成」](#)
- [135 ページの「プロフィールの管理」](#)
- [149 ページの「デスクトップからネットワーク構成を管理する」](#)

プロフィールを有効および無効にする

`netadm` コマンドを使用して、プロフィールのタイプや、プロフィールが固定であるかリアクティブであるかに関係なく、すべてのプロフィールを有効および無効にします。基本コマンドの構文は次のとおりです。

```
# netadm enable [ -p profile-type ] [ -c ncu-class ] profile-name
```

たとえば、次のようにして、システム定義の Automatic NCP を有効にします。

```
# netadm enable -p ncp Automatic
```

プロフィールの有効化と無効化に関する基本的な情報については、[113 ページの「プロフィールベースのネットワーク構成の使用ガイドライン」](#)を参照してください。

さまざまなタイプのプロフィールの有効化および無効化については、次の追加ガイドラインを参照してください。

- **NCP** – システムには常に、1 つのアクティブな NCP と 1 つのアクティブな場所プロフィールが必要です。アクティブな NCP は、別の NCP を明示的に有効にするまでアクティブなままです。別の NCP を有効にすると、現在アクティブな NCP は暗黙的に無効になります。システム上で現在アクティブな NCP を明示的に無効にすることはできません。108 ページの「[NCP の説明](#)」。

DefaultFixed NCP を有効にして固定モードに切り替えると、DefaultFixed の場所も自動的に有効になり、変更はできません。

Automatic NCP を有効にすると、アクティブ化ポリシーにより、現在のネットワーク状態に基づいて適切な該当の場所が選択され、有効になります。

- **NCU** – NCU のアクティブ化モードが `manual` に設定されている場合、現在アクティブな NCP の一部である個別の NCU を手動で有効および無効にできます。リンクまたは NCU クラスが指定されていない場合、両方の NCU が有効または無効になります。108 ページの「[NCU の説明](#)」を参照してください。
- **場所** – デフォルトでは、有効にする最適な場所プロフィールがシステムにより選択されます。システムは、`system` または `conditional` アクティブ化モードの場所セットから場所を選択します。ただし、場所の起動モードに関係なく、ユーザーはいつでも手動で場所を有効にしてシステムの選択をオーバーライドできます。場所を手動で有効にすると、システムはアクティブな場所を自動的に変更しません。場所の自動選択は無効になります。システムによる条件付きの場所選択に戻すには、手動で有効化された場所を明示的に無効にする必要があります。109 ページの「[場所プロフィールの説明](#)」を参照してください。
- **ENM** – これらのプロフィールは、`manual` または `conditional` アクティブ化モードに設定できます。`activation-mode` プロパティを `conditional` に設定した場合は、システムが指定された条件に基づいて ENM を有効または無効にします。アクティブ化モードを `manual` に設定した場合は、`netadm` コマンドを使用して ENM を有効または無効にできます。ENM のアクティブ化には制約はありません。常にシステムで 0 個以上の ENM をアクティブにすることができます。1 つの ENM を有効または無効にしても、ほかの現在アクティブな ENM には影響しません。114 ページの「[プロフィールのアクティブ化条件](#)」を参照してください。

例 6-1 NCP の有効化

次の例では、`myncp` という名前のユーザー定義 NCP が有効になります。

```
$ netadm enable -p ncp myncp
Enabling ncp 'myncp'
```


例 6-2 場所プロファイルの有効化

次の例では、office という名前の場所プロファイルが有効になります。

```
$ netadm enable -p loc office
Enabling loc 'office'
```

プロファイル名を指定する際に、netadm コマンドでは大文字と小文字が区別されないことに注意してください。

例 6-3 リンク NCU を無効にする

次の例では、net1 という名前のリンク NCU が無効になります。

```
$ netadm disable -p ncu -c phys net1
```

例 6-4 場所の無効化

次の例では、office という名前の場所が無効になります。

```
$ netadm disable -p loc office
Disabling loc 'office'
```

例 6-5 ENM の有効化と無効化

次の例では、ENM test-enm1 が有効になり、別の ENM test-enm2 が無効になります。

```
$ netadm enable -p enm test-enm1
Enabling enm 'test-enm1'
$ netadm disable -p enm test-enm2
Disabling enm 'test-enm2'
```

例 6-6 固定モードとリアクティブモードの切り替え

次の例では、システム定義の DefaultFixed NCP を有効にすることにより、固定モードに切り替える方法を示します。

```
$ netadm enable -p ncp DefaultFixed
Enabling ncp 'DefaultFixed'
```

プロファイルの構成

次の方法で netcfg コマンドを使用して、リアクティブプロファイルを構成できます。

- 対話型
- コマンド行モード
- コマンドファイルモード

netcfg 対話型モードでの操作

対話形式で使用するときは、netcfg コマンドで *scope* の概念を使用します。このコマンドを対話形式で使用する場合、スコープは常にプロファイルタイプおよび実行中の特定タスクに依存します。netcfg コマンドを単独で端末ウィンドウに入力する場合は、次の例に示すように、プロンプトが大域スコープに表示されます。

```
$ netcfg
netcfg>
```

プロファイルを作成または選択するには、最初にnetcfg 対話型セッションを開始する必要があります。

大域スコープのプロンプトから、select または create サブコマンドを使用すると、最上位のプロファイルである次のプロファイルタイプを表示、変更、または作成できます。

- NCP
- 場所
- ENM
- 既知の WLAN

プロファイルの作成後または選択後に、netcfg 対話型プロンプトは次の例のようになります。

```
netcfg:object-type:object-name>
```

netcfg コマンドを対話型モードで使用して、次のタスクを実行します。

- プロファイルを作成します。
- プロファイルを選択して変更します。
- プロファイルに関する必須情報がすべて設定され、有効であることを確認します。
- 新しいプロファイルの変更を確認します。
- 永続ストレージで変更を確定しないで、現在のプロファイル構成を取り消します。
- プロファイルに行なった変更を元に戻します。

場所プロファイルおよび ENM では、netcfg 対話型モードで最上位プロファイルを選択または作成すると、次の例に示すように、コマンドプロンプトがプロファイルスコープに表示されます。

```
netcfg> select loc test-loc
netcfg:loc:test-loc>
```

NCP が選択された場合は、コマンドプロンプトが *NCP スコープ* に表示されます。NCU は、このスコープで選択および作成されます。NCU を選択または作成すると、セッションが選択した NCP の *プロファイルスコープ* に移動します。このスコープでは、現在選択されているプロファイルに関連付けられたすべてのプロパティを表示および設定できます。

次の例では、office NCP が選択されており、これにより対話型セッションが NCP 用の NCP スコープに移動し、そこで NCU が選択されます。このアクションにより、選択された NCU のプロファイルスコープになります。このスコープでは、NCU のプロパティを表示または設定できます。

```
$ netcfg
netcfg> select ncp office
netcfg:ncp:office> select ncu phys net2
netcfg:ncp:office:ncu:net2>
```

どのスコープでも、コマンドプロンプトは現在選択されているプロファイルを示します。このスコープでプロファイルに行われた変更は、**確定** (つまり変更を永続ストレージに保存) できます。変更は、スコープの終了時に暗黙的に確定されます。変更を確定しない場合は、そのプロファイルを以前に確定したときの状態に戻すことができます。これにより、そのレベルでプロファイルに行なった変更が元に戻ります。revert サブコマンドと cancel サブコマンドは同じように動作します。

手順については、[124 ページの「NCP の作成」](#)を参照してください。

注記 - walkprop サブコマンドを使用して、プロファイルに関連付けられた各プロパティを個別に表示します。このコマンドは、対話型モードで使用する場合にのみ有効です。

netcfg コマンド行モードでの操作

コマンド行モードでは、選択したプロファイルまたはプロパティに影響を与える netcfg サブコマンドは、選択したプロファイルまたはプロパティが存在する特定のスコープ内で実行する必要があります。コマンド行モードでのプロファイルの選択は、対話型モードでの選択と同じです。唯一の違いは、コマンド行モードでは、すべてのサブコマンドがコマンド行上で書き込まれるという点です。たとえば、NCP の内容をリスト表示する場合、次の例に示すように、最初に NCP を選択してから、list サブコマンドを使用してその NCP の NCU を表示します。

```
$ netcfg "select ncp myncp; list"
ncp:myncp
```

```
management-type reactive
NCUs:
phys net0
phys net1
ip net0
ip net1
```

このモードで netcfg を使用する場合は、次のガイドラインを参照してください。

- 各サブコマンドをセミコロンで区切ります。
- select サブコマンドを大域スコープから指定して、NCP スコープに移動します。
- list サブコマンドを NCP スコープから指定して、そのスコープ内のプロパティをリスト表示します。
- 二重引用符を使用して、シェルでセミコロンが解釈されないようにします。

注記 - コマンド行モードでは、完全なコマンドを単一行で入力する必要があります。netcfg コマンドをコマンド行モードで使用することによって、選択したプロフィールに行なった変更は、コマンドの実行時に永続ストレージに確定されます。

walkprop サブコマンドを除くすべての netcfg サブコマンドを、コマンド行で使用できます。

netcfg コマンドファイルモードでの操作

コマンドファイルモードでは、プロフィール構成情報とコマンドがファイルから抽出されます。ファイル内のコマンドは、対話型モードおよび export サブコマンドで使用されるものと同じです。export サブコマンドと -f オプションを使用して、ファイルを生成します。たとえば、次のコマンドは現在のプロフィール構成をファイルにエクスポートします。

```
$ netcfg export -f /tmp/nwam.config
```

プロフィール構成をファイルからインポートするには、次のコマンドを入力します。

```
$ netcfg -f /tmp/nwam.config
```

export サブコマンドは、対話形式でも使用できます。詳細は、[146 ページの「プロフィール構成をエクスポートする」](#)を参照してください。

NCP の作成

NCP は、システムのネットワーク構成を定義します。固定 NCP ではなく、リアクティブ NCP だけを作成できます。

対話型モードで NCP を作成するには、まず対話型セッションを開始します。それから、次の例に示すように、`create` サブコマンドを使用して新しい NCP を作成します。

```
$ netcfg
netcfg> create ncp myncp
netcfg:ncp:myncp>
```

例 6-7 Automatic NCP をクローニングして NCP を作成する

DefaultFixed NCP 以外の既存の NCP をクローニングして、NCP を作成できます。次に、NCP のプロパティを変更して、新しい構成パラメータを指定します。次の例では、システム定義の Automatic NCP をクローニングすることによって、`newncp` という新しい NCP が作成されます。

```
$ netcfg
netcfg> create -t Automatic ncp newncp
netcfg:ncp:newncp> list
ncp:newncp
  management-type reactive
NCUs:
  phys net0
  phys net1
  ip net0
  ip net1
```

`management-type` プロパティは、常に `reactive` に設定されている読み取り専用のプロパティです。前の例では、`list` サブコマンドを使用して、新しくコピーされた NCP (`newncp`) の内容を表示します。

次の例に示すように、`netcfg` コマンド行モードを使用して、Automatic NCP を直接クローニングすることもできます。対話型とコマンド行のどちらの手法でも、既存の NCU を Automatic NCP から新たに作成された NCP にコピーします。

```
$ netcfg create -t Automatic ncp newncp
$ netcfg list ncp newncp
ncp:newncp
  management-type reactive
NCUs:
  phys net0
  phys net1
  ip net0
  ip net1
```

NCP に NCU を作成する

NCP は、基本的には、NCP のネットワーク構成を定義するプロパティで構成された NCU セットを含むコンテナです。すべての NCP には、リンクとインタフェースの両方の NCU が含まれています。

リンク NCU には、NCP のリンク構成とリンク選択ポリシーの両方を指定します。インタフェース NCU には、NCP のインタフェース構成ポリシーを指定します。IP 接続が必要な場合は、リンクとインタフェースの両方の NCU が必要です。netcfg コマンドまたはネットワーク管理 GUI を使用して、NCU を明示的に追加または削除する必要があります。ネットワーク管理 GUI を使用した NCU の追加および削除の詳細は、[149 ページの「デスクトップからネットワーク構成を管理する」](#)を参照してください。

netcfg コマンドを対話型またはコマンド行モードで使用して、NCU を作成します。NCU の作成には複数の操作が伴うため、NCU およびすべてのプロパティを作成する単一行のコマンドを構築しようとするよりも、対話的に使用した方が簡単かつ効率的に NCU を作成できます。NCU は、NCP の初期作成時または作成後に作成できます。NCU を作成または変更するプロセスには、一般的な NCU プロパティの設定に加えて、各 NCU タイプに特に適用されるプロパティの設定も含まれます。

NCU を対話形式で作成すると、netcfg コマンドによって NCU の関連する各プロパティが調査され、デフォルト値 (デフォルトが存在する場合) と指定可能なすべての値の両方が表示されます。たとえば、インタフェース NCU の `ipv4-addrsrc` プロパティに `dhcp` を指定する場合、このプロパティは静的 IP アドレス構成のみに使用されるため、`ipv4-addr` プロパティの値指定は求められません。対話型プロンプトで各プロパティの代替値を指定できます。値を指定しないで Return を押すと、デフォルト値が再適用されます (デフォルト値がない場合、プロパティは空白のままです)。

NCU の作成または変更時に指定可能な NCU プロパティがいくつかあります。プロパティの中には、両方の NCU タイプに適用されるものと、リンク NCU とインタフェース NCU のどちらかに適用されるものがあります。すべての NCU プロパティの詳細 (これらのプロパティの指定時に適用される可能性のある規則や条件を含む) は、[netcfg\(1M\)](#) のマニュアルページを参照してください。

▼ NCP 用の NCU を対話形式で作成する方法

次の手順は、既存の NCP を選択して、その NCP 用の NCU を対話形式で作成する方法を説明します。

注記 - プロファイルの初期作成中に実行される「調査」プロセスにより、NCP の変更時には、作成時に行われた選択に基づいて適用可能なプロパティに関するプロンプトのみが表示されます。

1. **netcfg 対話型セッションを開始します。**

```
$ netcfg
netcfg>
```

2. **既存の NCP を選択します。**

次の例では、NCP myncp が選択されています。

```
netcfg> select ncp myncp
netcfg:ncp:myncp>
```

NCP を選択すると、自動的に NCP のスコープに移動します。場所、ENM、または WLAN オブジェクトの場合は、コマンドプロンプトでそのプロファイルのプロファイルスコープに移動します。

3. **NCP にリンク NCU およびインタフェース NCU を作成します。**

次の例では、リンク NCU が作成されます。

```
netcfg:ncp:myncp> create ncu phys net0
Created ncu `net0'. Walking properties ...
activation-mode (manual) [manual|prioritized]>
mac-address>
autopush>
mtu> 1600
```

ここで ncu はオブジェクトタイプ、phys は NCU のクラス、net0 はオブジェクト名です。

NCU を作成すると、オブジェクトのスコープに移動し、オブジェクトのデフォルトプロパティが調査されます。

この例では、次のプロパティが指定されています。

- activation-mode プロパティのデフォルトは manual で、Return を押して受け入れます。
- mac-address および autopush プロパティは空のままになります。
- mtu プロパティは、値 1600 に設定されます。

次の例では、インタフェース NCU を作成する方法を示します。

```
netcfg:ncp:myncp> create ncu ip net0
Created ncu `net0'. Walking properties ...
ip-version (ipv4,ipv6) [ipv4|ipv6]> ipv4
ipv4-addrsrc (dhcp) [dhcp|static]> dhcp
ipv4-default-route>
```

ここで、ncu はオブジェクトタイプ、ip はオブジェクトクラス、net0 はオブジェクト名です。

NCU を作成すると、オブジェクトのスコープに移動し、オブジェクトのデフォルトプロパティが調査されます。

この例では、次のプロパティが指定されています。

- ip-version プロパティは ipv4 に設定されます。
- ipv4-addrsrc プロパティは dhcp に設定されます。

NCU の作成中は、2 つの NCU タイプを区別するために class オプションが使用されます。

4. (オプション) 次のようにして、構成が正しいことを確認します。

```
netcfg:ncp:myncp:ncu:net0> verify
All properties verified
```

verify サブコマンドは、構成を検証し、必要な値が見つからない場合は通知します。

5. 作成した各 NCU を保存します。

- commit サブコマンドを使用します。

```
netcfg:ncp:myncp:ncu:net0> commit
Committed changes
netcfg:ncp:myncp:ncu:net0>
```

commit サブコマンドは、暗黙的にプロパティを検証します。

- end サブコマンドを使用します。

```
netcfg:ncp:myncp:ncu:net0> end
Committed changes
netcfg:ncp:myncp>
```

end サブコマンドは、暗黙的に変更を確定します。

この場合、NCU の NCP への追加が完了すると、end サブコマンドによりセッションが NCP スコープに移動します。

対話型モードでは、確定されるまで変更は永続ストレージに保存されません。`commit` サブコマンドを使用すると、プロファイル全体が確定されます。永続ストレージの整合性を維持するため、`commit` 操作にも検証手順が含まれています。検証が失敗すると、`commit` 操作も失敗します。暗黙の確定処理に失敗すると、現在の変更を確定せずに対話型セッションを終了するオプションが与えられます。あるいは、現在のスコープに残って、プロファイルの変更を続行することもできます。

注記 - 行なった変更を取り消すには、`cancel` または `revert` サブコマンドを使用します。

`cancel` サブコマンドは、現在の変更を永続ストレージに確定せずに現在のプロファイル構成を終了します。これにより、対話型セッションは 1 つ上のレベルのスコープに移動します。`revert` サブコマンドは、行なった変更を元に戻し、以前の構成を再度読み込みます。`revert` サブコマンドを使用すると、対話型セッションは同じスコープ内にとどまります。

6. 完了したら、対話型セッションを終了します。

```
netcfg:ncp:myncp> exit
```

`exit` サブコマンドは `end` サブコマンドに似ていますが、対話型セッションの終了も実行します。

場所の作成

場所プロファイルには、基本リンクおよび IP 接続に直接関連しないネットワーク構成の値を定義するプロパティが含まれています。一部の例には、必要に応じて、同時に適用されるネームサービスおよび IP フィルタの設定が含まれています。正確に 1 つの場所プロファイルと 1 つの NCP がシステムに常に存在する必要があります。

`netcfg` コマンドを対話型またはコマンド行モードで使用して、場所プロファイルを作成します。場所プロファイルを作成する際は、その特定の場所を有効にする際に、特定の構成パラメータを定義する値を指定することによって、場所のプロパティを設定します。場所のプロパティは、特定クラスの構成設定を示すグループ別に分類されます。

場所のプロパティは、リポジトリにも格納されます。特定の場所が有効になると、そのプロパティが実行中のシステムに自動的に適用されます。場所プロファイルの作成および変更には、特定の場所をいつ有効にするかを定義するプロパティの設定も含まれます。構成プロセス中に表示されるプロパティは、以前に設定されたプロパティ値に基づいています。

すべての場所プロファイルのプロパティの詳細 (プロパティの指定時に適用される可能性のある規則、条件、依存関係など) は、[netcfg\(1M\)](#) のマニュアルページを参照してください。

▼ 場所を対話形式で作成する方法

次の手順では、場所プロファイルを作成する方法について説明します。プロファイルの初期作成中に実行される「調査」プロセスでは、事前に入力された値に基づいて適用可能なプロパティに関するプロンプトのみが表示されます。

1. netcfg 対話型セッションを開始します。

```
$ netcfg
netcfg>
```

2. 場所プロファイルを作成します。

次の例では、office という場所が作成されます。

```
netcfg> create loc office
Created loc 'office'. Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-any
conditions> ncu ip:net0 is active
nameservices (dns) [dns|files|nis|ldap]>
nameservices-config-file ("/etc/nsswitch.dns")>
dns-nameservice-configsrc (dhcp) [manual|dhcp]>
nfsv4-domain>
ipfilter-config-file> /export/home/test/wifi.ipf.conf
ipfilter-v6-config-file>
ipnat-config-file>
ippool-config-file>
ike-config-file> /etc/inet/ike/ikev1.config
ikev2-config-file>
ipsecpolicy-config-file>
```

場所を作成すると、その場所のプロファイルスコープに自動的に移動します。

この例では、次のプロパティが指定されました。

- activation-mode プロパティが conditional-any に設定されたため、アクティブ化の条件を指定できるコマンドプロンプトが表示されました。
- 場所のアクティブ化の条件は、ncu ip:net0 is active と指定されました。
- ipfilter-config-file プロパティには、/export/home/test/wifi.ipf.conf ファイルが指定されました。
- ike-config-file プロパティには、/etc/inet/ike/ikev1.config ファイルが指定されました。

- 残りのプロパティについては、Return を押してデフォルト値が受け入れられました。

3. (オプション) 次のように `list` サブコマンドを使用して、プロファイル構成を表示します。

```
netcfg:loc:office> list
loc:office
      activation-mode           conditional-any
      conditions                 "ncu ip:net0 is active"
      enabled                   false
      nameservices              dns
      nameservices-config-file  "/etc/nsswitch.dns"
      dns-nameservice-configsrc dhcp
      ipfilter-config-file      "/export/home/test/wifi.ipf.conf"
      ike-config-file           "/etc/inet/ike/ikev1.config"
```

4. プロファイル構成が正しいかどうかを検証します。

```
netcfg:loc:office> verify
All properties verified
```

`verify` サブコマンドは、構成を検証し、必要な値が見つからない場合は通知します。

5. 構成を確認したら、場所を保存します。

```
netcfg:loc:office> commit
Committed changes
```

また、`end` サブコマンドを使用してセッションを終了して、プロファイル構成を保存し、セッションを大域スコープに移動することもできます。

```
netcfg:loc:office> end
Committed changes
netcfg>
```

6. 対話型セッションを終了します。

```
netcfg> exit
```

ENM の作成

ENM では、VPN アプリケーションなどのアプリケーションまたはスクリプトが、NCP および場所プロファイルで指定された構成とは別 (外部) のネットワーク構成をいつ行うかを指定できます。ENM の詳細は、[110 ページの「ENM の説明」](#)を参照してください。

注記 - システムは、ENM を作成する可能性があるアプリケーションを自動的に認識しません。最初にこれらのアプリケーションをインストールして、システム上で構成してから、`netcfg` コマンドを使用して ENM を作成する必要があります。

ENM の作成時に指定可能なプロパティの詳細は、[netcfg\(1M\)](#) のマニュアルページを参照してください。

▼ ENM を対話形式で作成する方法

1. `netcfg` 対話型セッションを開始します。

```
$ netcfg
netcfg>
```

2. ENM を作成します。

```
netcfg> create enm test-enm
Created enm 'test-enm'. Walking properties ...
activation-mode (manual) [manual|conditional-any|conditional-all]>
fmri> svc:/application/test-enm:default
start>
stop>
netcfg:enm:test-enm>
```

ENM を作成すると、ENM のプロファイルスコープに自動的に移動し、各プロパティが調査されます。

この例では、`test-enm` ENM に次のプロパティが指定されます。

- `activation-mode` プロパティは手動に設定されており、Return を押して受け入れます。この値は `manual` に設定されているため、`conditions` プロパティを設定可能にすることはできません。
- `fmri` プロパティは `svc:/application/test-enm:default` に設定されます。
- この ENM では、`start` および `stop` プロパティは設定されません。

3. (オプション) プロファイル構成を表示します。

```
netcfg:enm:test-enm> list
enm:test-enm
activation-mode manual
enabled false
fmri "svc:/application/test-enm:default"
```

4. プロファイル構成が正しいかどうかを検証します。

```
netcfg:enm:test-enm> verify
All properties verified
```

verify サブコマンドは、構成を検証し、必要な値が見つからない場合は通知します。

5. ENM を保存します。

```
netcfg:enm:test-enm> commit
Committed changes
netcfg>
```

commit サブコマンドが、構成を検証して保存します。

また、end サブコマンドを使用してセッションを終了して、プロファイル構成を保存することもできます。

```
netcfg:enm:test-enm> end
Committed changes
```

6. 対話型セッションを終了します。

```
netcfg> exit
```

既知の WLAN を作成する

既知の WLAN は、システムに登録されている無線ネットワークの情報を格納するプロファイルです。NCP は、システムが接続されている各無線ネットワークから提供される構成情報に基づいて、無線インタフェースを自動的に構成できます。詳細は、[111 ページの「既知の WLAN の説明」](#)を参照してください。

WLAN の作成または変更時に指定可能なプロパティについては、[netcfg\(1M\)](#) のマニュアルページを参照してください。

▼ 既知の WLAN を対話形式で作成する方法

1. netcfg 対話型セッションを開始します。

```
$ netcfg
netcfg>
```

2. 既知の WLAN を作成します。

次の例では、ESSID という名前の無線ネットワークに接続する既知の WLAN を作成します。既知の WLAN は、無線ネットワーク名と同じ名前またはその ESSID にする必要があります。

```
netcfg> create wlan mywifi
Created wlan 'mywifi'. Walking properties ...
priority (0)> 100
bssids>
keyname> mywifi-key
keyslot>
security-mode [none|wep|wpa]> wpa
netcfg:wlan:mywifi>
```

WLAN を作成すると、WLAN のプロファイルスコープに自動的に移動し、各プロパティが調査されます。

この例では、既知の WLAN mywifi に次のプロパティが指定されます。

- priority プロパティの値は、デフォルト値の 0 から 100 に変更されます。
- keyname プロパティは mywifi-key に設定され、この無線ネットワークに対してセキュアなオブジェクトの名前が指定されます。詳細は、[157 ページの「セキュアな WiFi 通信の確立」](#)を参照してください。
- security-mode プロパティは wpa に設定されます。このプロパティは、この無線ネットワークで使用される暗号化のタイプを指定します。
- keyslot および bssid プロパティ値は、空のままになります。

3. (オプション) プロファイル構成を表示します。

```
netcfg:wlan:mywifi> list
known wlan:mywifi
  priority 100
  keyname "mywifi-key"
  security-mode wpa
netcfg:wlan:mywifi>
```

4. プロファイル構成が正しいかどうかを検証します。

```
netcfg:wlan:mywifi> verify
All properties verified
```

verify サブコマンドは、構成を検証し、必要な値が見つからない場合は通知します。

5. 既知の WLAN を保存します。

```
netcfg:wlan:mywifi> commit
Committed changes
```

commit サブコマンドが、構成を検証して保存します。

また、end サブコマンドを使用してセッションを終了して、プロファイル構成を保存し、セッションを大域スコープに移動することもできます。

```
netcfg:wlan:mywifi> end
Committed changes
netcfg>
```

6. 対話型セッションを終了します。

```
netcfg> exit
```

プロファイルの管理

このセクションには、次のトピックが含まれます。

- [135 ページの「プロファイルのプロパティ値を設定する」](#)
- [137 ページの「プロファイル構成に関する情報を取得する」](#)
- [141 ページの「walkprop サブコマンドを使用してプロファイルのプロパティ値を設定する」](#)
- [143 ページの「プロファイルに関する情報を表示する」](#)
- [145 ページの「プロファイルを削除する」](#)
- [146 ページの「プロファイル構成をエクスポートする」](#)
- [148 ページの「エクスポートされたプロファイル構成を復元する」](#)

プロファイルのプロパティ値を設定する

set サブコマンドを使用して、リアクティブプロファイルのプロパティ値が設定または変更されます。このサブコマンドは、対話形式またはコマンド行モードで使用できます。プロパティ値をコマンド行モードで設定または変更する場合は、変更が永続ストレージでただちに確定されます。

注記 - set サブコマンドを使用して、DefaultFixed NCP または DefaultFixed 場所プロファイルを変更することはできません。DefaultFixed NCP がアクティブな場合、常に dladm および ipadm コマンドを使用して構成を変更します。DefaultFixed の場所がアクティブな場合は、svccfg および svcadm コマンドを使用して、関連する SMF プロパティを直接変更します。[svccfg\(1M\)](#) および [svcadm\(1M\)](#) のマニュアルページを参照してください。

set サブコマンドの構文は、次のとおりです。

```
netcfg> set prop-name=value1[,value2,...]
```

▼ プロファイルプロパティ値を対話形式で設定する方法

次の手順では、場所プロファイルのプロパティ値を対話形式で設定する方法について説明します。プロパティ値を対話形式で設定する場合は、まず現在のスコープでプロファイルを選択します。これにより、対話型セッションがプロファイルのスコープに移動します。その後、選択したプロファイルが永続ストレージからメモリーに読み込まれます。これで、このスコープでプロファイルのプロパティを変更できます。

次の手順は、例を示すだけのため、`test-loc` の場所の `ipfilter-config-file` プロパティを設定する方法を示します。

1. **netcfg 対話型セッションを開始します。**

```
$ netcfg
netcfg>
```

2. **変更するプロファイルまたは構成オブジェクトを選択します。**

```
netcfg> select loc test-loc
netcfg:loc:test-loc>
```

3. **プロパティ値を設定します。**

次の例では、`ipfilter-config-file` プロパティが設定されます。

```
netcfg:loc:test-loc> set ipfilter-config-file = /path/to/ipf-file
```

4. **(オプション) 構成情報をリスト表示します。**

```
netcfg:loc:test-loc> list
loc:test-loc
activation-mode  manual
enabled          false
nameservices    dns
dns-nameservice-configsrc  dhcp
nameservices-config-file  "/etc/nsswitch.dns"
ipfilter-config-file  "/path/to/ipf-file"
```

5. **セッションを終了します。**

```
netcfg:loc:test-loc> end
Committed changes
netcfg>
```


end サブコマンドは、セッションを保存して大域スコープに移動します。

6. 対話型セッションを終了します。

```
netcfg> exit
```

例 6-8 コマンド行モードでプロファイルのプロパティ値を設定する

前の例では ipfilter-config-file プロパティを対話形式で設定する方法を示しましたが、同じことを次のようにコマンド行モードで実行することもできます。

```
$ netcfg "select loc test-loc; set ipfilter-config-file = /path/to/ipf-file"
```

コマンド行モードは、単純なアクションのみを実行する必要がある場合に最適です。ただし、コマンド行モードを使用し、適切なサブコマンドを注意深くコマンド行に指定することで、より複雑なアクションを実行することもできます。対話形式の例と同様に、コマンド行モードでも、最初に場所を選択して、そのプロファイルのスコープに移動する必要があります。次に set サブコマンドを指定して、個別のプロパティ値を設定します。

コマンド行モードを使用するときは、特定のプロパティに同時に複数の値を設定できます。この方法で複数の値を設定する場合は、各値をコンマ (,) で区切る必要があります。指定されたプロパティの各値にコンマも含まれている場合は、プロパティ値の一部であるコンマの前に、バックスラッシュ (\) を付ける必要があります。単一の値のみが含まれるプロパティ内のコンマは、区切り文字とは解釈されないため、前にバックスラッシュを付ける必要はありません。

たとえば、次のように ip-version プロパティを設定して、IPv4 と IPv6 の両方を myncp NCP 内の net0 NCU で使用します。

```
$ netcfg "select ncp myncp; select ncu ip net0; set ip-version=ipv4,ipv6"
```

プロファイル構成に関する情報を取得する

対話形式またはコマンド行モードで list サブコマンドを使用して、すべてのプロファイル、プロパティと値のペア、および現在のまたは指定したスコープに存在するリソースをリスト表示します。

次の例に示すように、大域スコープでは、list サブコマンドは、システム上のすべてのシステム定義およびユーザー定義プロファイルをリスト表示します。

```
$ netcfg list
netcfg list
NCPs:
  DefaultFixed
  Automatic
  myncp
Locations:
  Automatic
  NoNet
  DefaultFixed
  office
ENMs:
  test-enm
WLANs:
  mywifi
```

注記 - 対話型モードでは、大域スコープ内で `list` サブコマンドを使用して、同じ情報をリスト表示します。

`list` サブコマンドでは、各プロファイルの状態はリスト表示されません。プロファイルとその状態に関する情報を表示するには、`netadm` コマンドと `list` サブコマンドを使用します。詳細は、[144 ページの「プロファイルの現在の状態を表示する」](#)を参照してください。

個別のプロファイルのプロパティ値をリスト表示する

プロファイルスコープ内の `list` サブコマンドは、指定されたプロファイルのすべてのプロパティ値をリスト表示します。構文は次のとおりです。

```
netcfg> list [ object-type [ class ] object-name ]
```

▼ プロファイルのすべてのプロパティ値を対話形式でリスト表示する方法

次の手順では、`list` サブコマンドを対話形式で使用して、プロファイルのすべてのプロパティ値をリスト表示する方法を説明します。次の手順の例では、IP NCP 名 `net0` の構成情報をリスト表示する方法を示します。リスト表示される各プロファイルの値は、プロファイルごとに異なります。

1. **netcfg** 対話型セッションを開始します。

```
$ netcfg
netcfg>
```

2. **NCP** を選択します。

```
netcfg> select ncp myncp
netcfg:ncp:myncp>
```

3. 次のいずれかの方法で、IP NCU の構成をリスト表示します。

■ 大域スコープから構成をリスト表示します。

```
netcfg:ncp:myncp> list ncu ip net0
ncu:net0
      type           interface
      class          ip
      parent         "myncp"
      enabled        false
      ip-version     ipv4,ipv6
      ipv4-addrsrc   dhcp
      ipv6-addrsrc   dhcp,autoconf
netcfg:ncp:myncp>
```

■ プロファイルスコープから構成をリスト表示します。

```
netcfg:ncp:myncp> select ncu ip net0
netcfg:ncp:myncp:ncu:net0> list
ncu:net0
      type           interface
      class          ip
      parent         "myncp"
      enabled        false
      ip-version     ipv4,ipv6
      ipv4-addrsrc   dhcp
      ipv6-addrsrc   dhcp,autoconf
netcfg:ncp:myncp:ncu:net0>
```

4. 対話型セッションを終了します。

```
netcfg:ncp:myncp:ncu:net0> exit
```

例 6-9 コマンド行モードでプロパティ値をリスト表示する

前の例ではプロパティ値を対話形式でリスト表示する方法を示しましたが、同じことをコマンド行モードでも実行できます。使用するモードに関係なく、出力は同じになります。

たとえば、次のようにして NCP スコープから net0 IP NCU のプロパティをリスト表示します。

```
$ netcfg "select ncp myncp; list ncu ip net0"
```

プロファイルスコープから net0 IP NCU のプロパティをリスト表示するには、次のコマンドを使用します。

```
$ netcfg "select ncp myncp; select ncu ip net0; list"
```

プロフィールの特定のプロパティ値を取得する

`get` サブコマンドを使用して、プロフィールの特定のプロパティ値を取得します。このサブコマンドは、対話形式またはコマンド行モードで使用できます。このコマンドの構文は次のとおりです。

```
netcfg> get [ -V ] prop-name
```

▼ プロフィールの特定のプロパティ値を取得する方法

次の手順では、`get` サブコマンドを対話形式で使用して、プロフィールの特定のプロパティ値を取得する方法を説明します。次の手順の例では、IP NCU の `ip-version` プロパティ値を取得する方法を示します。

1. `netcfg` 対話型セッションを開始します。

```
$ netcfg
netcfg>
```

2. NCP を選択してから、IP NCU を選択します。例:

```
netcfg> select ncp myncp
netcfg:ncp:myncp> select ncu ip net0
netcfg:ncp:myncp:ncu:net0>
```

3. 次のコマンドのいずれかを使用して、特定のプロパティ値を取得します。

- 次のように `get` サブコマンドを使用して、プロパティ名およびプロパティ値を表示します。

```
netcfg:ncp:myncp:ncu:net0> get ip-version
ip-version ipv4,ipv6
```

- プロパティ名を表示せずに、プロパティ値のみを取得する場合は、次のように `get` サブコマンドと `-V` オプションを使用します。

```
netcfg:ncp:myncp:ncu:net0> get -V ip-version
ipv4,ipv6
```

4. 対話型セッションを終了します。

```
netcfg:ncp:myncp:ncu:net0> exit
```

例 6-10 コマンド行モードでプロファイルの特定のプロパティ値を取得する

前の対話形式の例は、コマンド行モードでも実行できます。使用するモードに関係なく、出力は同じになります。

たとえば、次のようにして、コマンド行モードで IP NCU の `ip-version` プロパティの値を取得します。

```
$ netcfg "select ncp myncp; select ncu ip net0; get ip-version"
ip-version ipv4,ipv6
```

次の例では、`get` サブコマンドと `-v` オプションを使用して特定のプロパティ値を取得する方法を示します。この方法は、プロパティ名を解析する必要がないスクリプトで役立ちます。

```
$ netcfg "select ncp myncp; select ncu ip net0; get -V ip-version"
ipv4,ipv6
```

walkprop サブコマンドを使用してプロファイルのプロパティ値を設定する

`walkprop` サブコマンドを使用して、プロファイルの個別のプロパティ値を対話形式で表示および変更します。対話形式のセッションを開始してから `walkprop` サブコマンドを入力すると、プロファイルの各プロパティの名前および現在値を、一度に 1 つのプロパティずつ表示できます。さまざまなプロパティを表示するとき、必要に応じて現在値やデフォルト値を設定または変更できます。

注記 - `walkprop` サブコマンドは、対話型モードでのみ使用することを意図しています。

▼ 指定されたプロファイルのプロパティ値を「調査」および設定する方法

次の手順では、`walkprop` サブコマンドを使用して、指定されたプロファイルのプロパティ値を対話形式で表示および変更する方法を説明します。次の例に示すように、`walkprop` サブコマンドを使用してプロファイルのプロパティを設定する場合、`set` サブコマンドを使用する必要はありません。

1. `netcfg` 対話型セッションを開始します。

```
$ netcfg
netcfg>
```

2. プロパティを表示および変更するプロファイルを選択します。

次の例では、test-loc という名前の場所が選択されます。

```
netcfg> select loc test-loc
netcfg:loc:test-loc>
```

3. **walkprop** サブコマンドを入力して調査を開始します。

次の例では、walkprop サブコマンドの発行後に表示される最初のプロパティは、activation-mode プロパティです。現在、プロパティのデフォルト値は manual (カッコ内に表示) です。

```
netcfg:loc:test-loc> walkprop
activation-mode (manual) [manual|conditional-any|conditional-all]>
```

4. プロパティの値を変更するには、対話型プロンプトで新しい値を入力して、Return を押し

ます。
たとえば、次のようにして、場所の activation-mode プロパティを manual から conditional-all に変更します。

```
netcfg:loc:test-loc> walkprop
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-all
```

Return を押すと、現在の設定を保存して、次のプロパティを調査します。

5. プロファイルのすべてのプロパティが表示されるまで、調査プロセスを繰り返し、手順 4 の手順に従い、必要に応じて変更を加えます。

```
netcfg:loc:test-loc> walkprop
activation-mode (manual) [manual|conditional-any|conditional-all]> conditional-all
conditions> advertised-domain is example.com
nameservices (dns) [dns|files|nis|ldap]>
nameservices-config-file ("/etc/nsswitch.dns")>
dns-nameservice-configsrc (dhcp) [manual|dhcp]>
nfsv4-domain>
ipfilter-config-file>
ipfilter-v6-config-file>
ipnat-config-file>
ippool-config-file>
ike-config-file>
ikev2-config-file>
ipsecpolicy-config-file>
```

プロパティを変更せずに Return を押すと、既存のデフォルト値が保持され、次のプロパティの調査に進みます。

注記 - 130 ページの「場所を対話形式で作成する方法」に記載されているように、指定されたプロファイルに関連するプロパティだけが表示されます。

6. プロファイルの現在のデフォルトプロパティ値をリスト表示します。例:

```
netcfg:loc:test-loc> list
loc:test-loc
activation-mode    conditional-all
conditions         "advertised-domain is example.com"
enabled            false
nameservices       dns
nameservices-config-file  "/etc/nsswitch.dns"
dns-nameservice-configsrc  dhcp
```

前の出力で、activation-mode プロパティが conditional-all に設定されていることを確認してください。

7. 対話型セッションを終了して、変更を確定します。

```
netcfg:loc:test-loc> exit
Committed changes
```

プロファイルに関する情報を表示する

netadm コマンドと list サブコマンドを使用して、各プロファイルの現在の状態を含む、システム上のいずれかまたはすべてのプロファイルの情報を表示します。

list サブコマンドの構文は、次のとおりです。

```
$ netadm list [ -p object-type ] [ -c ncu-class ] [ object-name ]
```

次のようにして、システムのすべてのプロファイル、および各プロファイルの現在の状態を表示します。

```
$ netadm list
TYPE  PROFILE  STATE
ncp   DefaultFixed  disabled
ncp   Automatic  online
ncu:phys  net0    online
ncu:phys  net1    offline
ncu:ip    net0    online
ncu:ip    net1    offline
loc   Automatic  online
loc   NoNet     offline
loc   DefaultFixed  offline
enm   test-enm  disabled
```

list サブコマンドでは、有効な NCP および特定の NCP を構成するすべての NCU が表示されます。

プロファイルの現在の状態を表示する

プロファイルタイプと NCU クラスを `list` サブコマンドに含めて、特定のプロファイルを識別できます。プロファイルタイプのみを指定した場合は、そのタイプのプロファイルがすべて表示されます。プロファイルを名前指定した場合は、そのプロファイルの現在の状態が表示されません。プロファイル名が一意でない場合は、その名前プロファイルがすべて一覧表示されます。

例 6-11 指定されたプロファイルの現在の状態を表示する

次の例では、`Automatic` という名前のシステムにあるすべてのプロファイルの現在の状態をリスト表示します。

```
$ netadm list Automatic
TYPE      PROFILE      STATE
ncp       Automatic    online
ncu:ip    net1         offline
ncu:phys  net1         offline
ncu:ip    net0         online
ncu:phys  net0         online
loc       Automatic    online
```

次の例では、`-p` オプションとともに `list` サブコマンドを使用して、システムに現在存在する場所をすべて表示します。前の例では、`Automatic` の場所はオンライン (有効) です。

```
$ netadm list -p loc
TYPE PROFILE STATE
loc DefaultFixed offline
loc NoNet offline
loc Automatic online
```

次の例では、`-c` オプションとともに `list` サブコマンドを使用して、アクティブな NCP のインタフェース NCU をすべて表示します。

```
$ netadm list -c ip
TYPE      PROFILE      STATE
ncu:ip    net0         online
ncu:ip    net1         offline
```

プロファイルの補助的な状態の値を表示する

プロファイルの補助的な状態は、指定されたプロファイルがオンラインまたはオフライン (有効または無効) である原因について説明しています。補助的な状態の値をリスト表示するには、次のように `-x` オプションを `list` サブコマンドで使用します。


```
$ netadm list -x
TYPE          PROFILE          STATE          AUXILIARY STATE
ncp           DefaultFixed    disabled      disabled by administrator
ncp           Automatic        online         active
ncu:phys      net0             online         interface/link is up
ncu:phys      net1             offline        interface/link is down
ncu:ip         net0             online         interface/link is up
ncu:ip         net1             offline        conditions for activation are unmet
loc           Automatic        offline        conditions for activation are unmet
loc           NoNet            offline        conditions for activation are unmet
enm test-enm      disabled      disabled by administrator
loc           DefaultFixed    offline        conditions for activation are unmet
```

補助的な状態の値は、プロフィールタイプによって異なります。補助的な状態の詳細は、[nwapd\(1M\)](#) のマニュアルページを参照してください。

プロフィールを削除する

`destroy` サブコマンドを使用して、ユーザー定義プロフィールおよび NCU などの構成オブジェクトを削除します。システム定義のプロフィールを削除することはできません (Automatic や DefaultFixed NCP、および Automatic、NoNet、DefaultFixed の各場所など)。現在アクティブなプロフィールを削除することはできません。最初にプロフィールを無効にしてから、削除する必要があります。

`destroy` サブコマンドの構文は、次のとおりです。

```
netcfg> destroy [ -a | object-type [ class ] object-name ]
```

`-a` オプションは、現在アクティブなユーザー定義プロフィールを除く、すべてのユーザー定義プロフィールをシステムから削除します。

▼ プロフィールを対話形式で削除する方法

次の手順では、ユーザー定義のプロフィールを対話形式で削除する方法について説明します。この手順は、例を示すだけのため、`myncp` ユーザー定義 NCP から IP NCU を削除する方法を示します。

1. `netcfg` 対話型セッションを開始します。

```
$ netcfg
netcfg>
```

2. プロファイルを選択します。

たとえば、myncp NCP を選択するには、次のコマンドを入力します。

```
netcfg> select ncp myncp
netcfg:ncp:myncp>
```

3. プロファイルまたは構成オブジェクトを削除します。

次の例では、net1 IP NCU が myncp NCP から削除されます。

```
netcfg:ncp:myncp> destroy ncu ip net1
netcfg:ncp:myncp>
```

4. 対話型セッションを終了します。例:

```
netcfg:ncp:myncp> exit
```

例 6-12 コマンド行モードでプロフィールを削除する

前の例では、プロフィールを対話形式で削除する方法を示しましたが、同じことを次のようにコマンド行モードで実行できます。

```
$ netcfg "select ncp myncp; destroy ncu ip net1"
```

プロフィール構成をエクスポートする

export サブコマンドを使用して、プロフィール構成を保存します。プロフィールのエクスポートは、同一のネットワーク構成が必要な複数のサーバーの保守担当者にとって役立つことがあります。export サブコマンドは、対話形式でもコマンド行モードでも使用できます。プロフィールのエクスポート時に、出力が netcfg コマンドにより解釈可能な一連のサブコマンドとして表示されます。これらのサブコマンドは、対話型モードまたはコマンド行モードで入力するコマンドに似ています。

注記 - 構成によっては、export 機能の使用が制限されます。エクスポートできるのは、最初に netcfg コマンドを使用して作成された構成オブジェクトだけです。dladm または ipadm コマンドを使用して作成された構成オブジェクト (リンクアグリゲーションや IPMP グループなど) はエクスポートできません。また、DefaultFixed NCP および DefaultFixed の場所もエクスポートできません。

export サブコマンドの構文は、次のとおりです。

```
netcfg> export [ -d ] [ -f output-file ] [ object-type [ class ] object-name ]
```

注記 - export サブコマンドの -d および -f オプションは、相互に独立して使用できます。-f オプションは、現在のスコープまたは指定したスコープの現在の構成を指定したファイルに出力します。-d オプションは、出力の最初の行として `destroy -a` コマンドを追加します。

例 6-13 プロファイル構成を対話形式でエクスポートする

次の例は、`export` サブコマンドを対話形式で使用して、プロファイル構成を画面に表示する方法を示します。

```
$ netcfg
netcfg> export
create ncp "myncp"
create ncu ip "net0"
set ip-version=ipv4
set ipv4-addrsrc=dhcp
set ipv6-addrsrc=dhcp,autoconf
end
create ncu phys "net0"
set activation-mode>manual
set mtu=5000
end
end
create loc "test-loc"
set activation-mode=conditional-all
set conditions="system-domain is example.com"
set nameservices=dns
set nameservices-config-file="/etc/nsswitch.dns"
set dns-nameservice-configsrc=dhcp
end
create enm "test-enm"
set activation-mode=conditional-all
set conditions="ip-address is-not-in-range 10.2.3.4"
set fmri="svc:/application/test-enm:default"
end
create wlan "mywifi"
set priority=100
set keyname="mywifi-key"
set security-mode=wpa
end
```

コマンド行モードで、次のコマンドを入力します。

```
$ netcfg export
```

次の例に示すように (簡潔にするために切り詰められています)、-d オプションと `export` サブコマンドを使用して、`destroy -a` コマンドを `netcfg export` 出力の最初の行として追加できます。

```
$ netcfg
netcfg> export -d
```

```
destroy -a
create ncp "myncp"
create ncu ip "net0"
set ip-version=ipv4
set ipv4-addrsrc=dhcp
.
.
.
```

コマンド行モードで、次のコマンドを入力します。

```
$ netcfg export -d
```

例 6-14 プロファイル構成をファイルにエクスポートする

次の例では、`export` サブコマンドと `-f` オプションを使用して、`myncp` NCP の構成情報がファイルに書き込まれます。次の例では、`-f` オプションにより、出力が `myncp2` という名前の新しいファイルに書き込まれます。`-d` オプションを使用すると、`destroy -a` コマンドが `netcfg export` 出力の先頭行として追加されます。

次のようにして、プロフィール構成を対話形式でファイルにエクスポートします。

```
$ netcfg
netcfg> export -d -f myncp2
```

次に示すように、同じタスクをコマンド行モードで実行します。

```
$ netcfg export -d -f myncp2
```

次の短縮された例は、プロフィール構成を表示する方法を示しています。

```
$ cat myncp2
destroy -a
create ncp "myncp"
create ncu ip "net0"
.
.
.
```

エクスポートされたプロフィール構成を復元する

`export` サブコマンドで生成されたプロフィール構成をシステムに再度インポートできます。エクスポートされたファイルには、`netcfg` コマンドが解釈可能な一連のサブコマンドが存在します。`export` サブコマンドは、プロフィール構成を復元したり、別のシステムにコピーしたりする必

要がある場合に役立ちます。プロファイル構成を復元できるのは、`netcfg` コマンド行モードを使用する場合だけです。

次のコマンドを使用して、エクスポートした構成を復元します。

```
$ netcfg -f file
```

たとえば、次のようにして、[例6-14「プロファイル構成をファイルにエクスポートする」](#)でエクスポートしたファイルを復元します。

```
$ netcfg -f myncp2  
Configuration read.
```

デスクトップからネットワーク構成を管理する

ネットワーク管理 GUI (以前の NWAN) を使用して、デスクトップからネットワーク構成を管理できます。このツールは、`netcfg` および `netadm` コマンドを使用する場合と似ています。この GUI を使用して、有線または無線ネットワークに接続したり、新しい有線または無線接続を構成したり、Location プロファイルを作成したり、プロファイルを有効または無効にしたりできます。デスクトップからのネットワーク構成の管理は、ノートブック PC のユーザーや、工作中にホームオフィスから無線ネットワークへ切り替えるときや旅行中など、ネットワーク状態が頻繁に変わる場合にもっとも有効です。

デスクトップからの無線ネットワークの管理に関するタスク関連情報は、[160 ページの「デスクトップから無線ネットワークを管理する」](#)を参照してください。

デスクトップからネットワーク構成を管理する場合は、これらの一般的なガイドラインおよびベストプラクティスに従います。

- デスクトップからネットワーク構成を管理する場合、もっとも簡単な解決方法は、システム定義の Automatic NCP を有効にするというものです。自宅では、この NCP を使用して、無線ネットワークに接続できます。
- 有線接続を使用することにした場合は、Ethernet ケーブルを差し込みます。デフォルトの Automatic NCP から別の NCP に切り替えしないでください。ネットワーク接続は、無線ネットワーク接続から有線ネットワーク接続に自動的に適応し、既存のネットワーク構成にその他の変更を加える必要はありません。
- オフィスでも同じ規則が適用されます。Ethernet ケーブルがネットワークに接続されておらず、Automatic NCP が有効になっている場合、リアクティブネットワークが使用され、無線ネットワーク接続が自動的に確立されます。

- DefaultFixed NCP が現在アクティブな場合は、そのステータスのみを表示できます。この NCP がアクティブな場合にネットワークを構成するには、`dladm` および `ipadm` コマンドを使用する必要があります。第2章「Oracle Solaris でのデータリンク構成の管理」および第3章「Oracle Solaris での IP インタフェースとアドレスの構成および管理」を参照してください。
- 自宅とオフィスのどちらのシナリオでも、まだ行なっていない場合は最初に無線ネットワークを選択し、それをお気に入りの無線ネットワークのリストに保存する必要があります。
ネットワーク管理 GUI を使用するか、`netadm select-wifi` コマンドを実行することによって、無線ネットワークを選択します。159 ページの「リアクティブモードでの既知の WLAN の管理」を参照してください。
- 現在のネットワーク接続のステータスを表示するには、デスクトップ上に置かれている「ネットワークステータス」通知アイコンの上にマウスを置くか、単にアイコンをクリックします。「ネットワークステータス」通知アイコンには、GUI でネットワーク構成を作成および管理するためのコンテキストメニューも含まれます。
「ネットワークステータス」通知アイコンがデスクトップに表示されていない場合は、「システム」->「管理」->「ネットワーク」の順に選択して開始します。コマンド行から GUI を開始するには、`nwam-manager` コマンドを実行します。詳細は、JDS/GNOME マニュアルページコレクション内の `nwam-manager(1M)` のマニュアルページを参照してください。
- IP 関連の構成は、「ネットワーク設定」ダイアログボックスの「ネットワークプロファイル」セクションで管理されます。デスクトップ上に置かれた「ネットワークステータス」通知アイコンをクリックするか、「ネットワークステータス」通知アイコンのコンテキストメニューから「ネットワーク設定」オプションを選択して、「ネットワーク設定」ダイアログボックスにアクセスします。

◆◆◆ 第 7 章

Oracle Solaris での無線ネットワークの管理

この章には、Oracle Solaris リリースでの無線ネットワークの管理タスクが含まれます。

IEEE 802.11 仕様では、ローカルエリアネットワークでの無線通信が定義されています。これらの仕様と、そこに記載されているネットワークはまとめて *WiFi* と呼ばれます。この用語は、Wi-Fi Alliance 業界グループによって商標登録されています。WiFi ネットワークは、プロバイダと想定クライアントの両方が比較的容易に構成できます。そのため、WiFi ネットワークはますます普及し、世界中で一般的に使用されています。WiFi ネットワークでは、携帯電話、テレビ、およびラジオと同じ電波テクノロジーが使用されます。

注記 - Oracle Solaris には、WiFi サーバーまたはアクセスポイントを構成するための機能は含まれていません。

この章の内容は、次のとおりです。

- [151 ページの「コマンド行を使用した無線ネットワークの管理」](#)
- [157 ページの「セキュアな WiFi 通信の確立」](#)
- [159 ページの「リアクティブモードでの既知の WLAN の管理」](#)
- [160 ページの「デスクトップから無線ネットワークを管理する」](#)

コマンド行を使用した無線ネットワークの管理

このセクションでは次のタスクについて説明します。

- [151 ページの「WiFi ネットワークに接続する方法」](#)
- [156 ページの「WiFi リンクをモニターする方法」](#)

▼ WiFi ネットワークに接続する方法

始める前に 次の手順を実行して、ノートブック PC を WiFi ネットワークに接続します。

1. 管理者になります。
2. システムに存在するデータリンクの物理属性を表示します。

```
# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX  DEVICE
net0          Ethernet      up      1500   full   ath0
net1          Ethernet      up      1500   full   e1000g0
```

前の例では、2 つのリンクが使用可能であることが出力に示されています。デバイス `ath0` リンク上の `net0` は、WiFi 通信をサポートします。`e1000g0` リンクを使用すると、システムを有線ネットワークに接続できます。

3. WiFi インタフェースを構成します。
 - a. WiFi をサポートするインタフェースを作成します。例:

```
# ipadm create-ip net0
```

- b. リンクが `plumb` されていることを確認します。

```
# ipadm show-if
IFNAME      CLASS      STATE    ACTIVE  OVER
lo0         loopback  ok       yes     --
net0       ip         ok       yes     --
```

4. 使用可能なネットワークを確認します。

```
# dladm scan-wifi
LINK      ESSID      BSSID/IBSSID  SEC    STRENGTH  MODE  SPEED
net0     ofc        00:0e:38:49:01:d0  none  good      g     54Mb
net0     home      00:0e:38:49:02:f0  none  very weak g     54Mb
net0     linksys   00:0d:ed:a5:47:e0  none  very good g     54Mb
```

`scan-wifi` コマンドは、現在の場所で使用可能な WiFi ネットワークに関する情報を表示します。出力には次の情報が含まれます。

LINK	WiFi 接続で使用されるリンク名を示します。
ESSID	拡張サービスセット識別子を示します。ESSID は WiFi ネットワークの名前であり、特定の無線ネットワークの管理者が任意に設定できます。
BSSID/IBSSID	特定の ESSID の一意の識別子である基本サービスセット識別子 (BSSID) を示します。BSSID は、ネットワークに特定の ESSID を提供する、隣接するアクセスポイントの 48 ビットの MAC アドレスです。

SEC	無線ネットワークへのアクセスに必要なセキュリティのタイプを示します。値は、none、WEP、および WPA です。詳細は、 157 ページの「セキュアな WiFi 通信の確立」 を参照してください。
STRENGTH	現在の場所で使用可能な WiFi ネットワークからの無線信号の強度を示します。
MODE	ネットワークによって実行される 802.11 プロトコルのバージョンを示します。モードは、a、b、g か、あるいはこれらのモードの任意の組み合わせです。
SPEED	特定のネットワークの速度 (メガビット/秒) を示します。

5. 次のいずれかの方法を使用して、WiFi ネットワークに接続します。

- 信号強度のもっとも高い、セキュリティ保護されていない WiFi ネットワークに接続します。

```
# dladm connect-wifi
```

- ESSID を指定することによって、セキュリティ保護されていないネットワークに接続します。

```
# dladm connect-wifi -e ESSID
```

dladm connect-wifi コマンドの使用に関する詳細は、[157 ページの「セキュアな WiFi 通信の確立」](#)および[dladm\(1M\)](#)のマニュアルページを参照してください。

6. 次のようにして、システムの接続先の WiFi ネットワークのステータスを確認します。

```
# dladm show-wifi
LINK      STATUS      ESSID      SEC      STRENGTH  MODE  SPEED
net0      connected   ofc        none     very good g      36Mb
```

前の出力は、システムが ofc ネットワークに接続されていることを示しています。この手順の手順 4 の scan-wifi の出力は、使用可能なネットワークで ofc の信号強度がもっとも高いことを示していました。別の無線ネットワークを明示的に指定しないかぎり、dladm connect-wifi コマンドは、信号強度のもっとも高い WiFi ネットワークを自動的に選択します。

7. 次のいずれかの方法を使用して、インタフェースの IP アドレスを構成します。

- DHCP サーバーから IP アドレスを取得します。

```
# ipadm create-addr -T dhcp interface
```

WiFi ネットワークで DHCP がサポートされていない場合は、次のメッセージが表示されま
す。

```
ipadm: interface: interface does not exist or cannot be managed using DHCP
```

■ 静的 IP アドレスを構成します。

```
# ipadm create-addr -a address interface
```

このオプションは、システムに専用の IP アドレスがある場合に使用します。

8. 次のいずれかの方法で、WiFi ネットワーク経由でインターネットにアクセスします。

■ アクセスポイントが無料のサービスを提供している場合は、ブラウザまたは任意のアプリ
ケーションを実行できます。

■ アクセスポイントが料金を必要とする商用 WiFi ネットワーク内にある場合は、その場所で
提供されている手順に従います。

通常、このオプションでは鍵および支払い方法を指定する必要があります。

9. 次のいずれかの方法でセッションを終了します。

■ WiFi セッションを終了しますが、システムは実行中のままにします。

```
# dladm disconnect-wifi
```

■ 現在、複数のセッションが実行されている場合は、特定の WiFi セッションを終了します。

```
# dladm disconnect-wifi link
```

ここで、*link* はセッションに使用されているインタフェースを表します。

■ WiFi セッションの実行中に、システムを正常にシャットダウンします。

```
# shutdown -g0 -i5
```

システムをシャットダウンする前に WiFi セッションを明示的に切断する必要はありません。

例 7-1 特定の WiFi ネットワークへの接続

次の例は、Oracle Solaris システムを無線ネットワークに接続するときに行う複数の異なる手順を組み合わせたものです。この例は、OS で任意の無線ネットワークを選択する代わりに、希望する特定の無線ネットワークにシステムを強制的に接続する方法も示しています。次の例では、ノートブック PC で使用する静的 IP アドレスとして 10.192.16.3/24 が割り当てられているものとします。

```
# dladm show-phys
LINK          MEDIA          STATE  SPEED  DUPLEX  DEVICE
net0          Ethernet      up     1500   full    ath0
net1          Ethernet      up     1500   full    e1000g0

# ipadm create-ip net0
# ipadm show-if net0
IFNAME      CLASS      STATE  ACTIVE  OVER
lo0         loopback  ok     yes     --
net0        ip         ok     yes     --

# dladm scan-wifi
LINK      ESSID      BSSID/IBSSID  SEC  STRENGTH  MODE  SPEED
net0      wifi-a     00:0e:38:49:01:d0  none  weak      g     54Mb
net0      wifi-b     00:0e:38:49:02:f0  none  very weak g     54Mb
net0      ofc-net    00:0d:ed:a5:47:e0  wep   very good g     54Mb
net0      citinet    00:40:96:2a:56:b5  none  good      b     11Mb

# dladm connect-wifi -e citinet
# dladm show-wifi
LINK      STATUS      ESSID      SEC  STRENGTH  MODE  SPEED
net0      connected   citinet    none  good      g     11Mb

# ipadm create-addr -a 10.192.16.3/24 net0
ipadm: net0/v4
# ipadm show-addr net0
ADDROBJ  TYPE      STATE      ADDR
net0/v4  static   ok         10.192.16.3/24
```

ブラウザまたは別のアプリケーションを開始して、WiFi ネットワーク経由で作業を開始します。

```
# firefox
```

セッションを終了しますが、PC は実行中のままにします。

```
# dladm disconnect-wifi
# dladm show-wifi
LINK      STATUS      ESSID      SEC  STRENGTH  MODE  SPEED
net0      disconnected  --         --   --        --   --
```

show-wifi の出力によって、net0 リンクが WiFi ネットワークから切り離されたことを確認できます。

▼ WiFi リンクをモニターする方法

次の手順では、標準のネットワークツールを使用して WiFi リンクのステータスをモニタリングする方法、および選択したリンクプロパティを変更する方法について説明します。

1. 管理者になります。
2. [151 ページの「WiFi ネットワークに接続する方法」](#)の説明に従って、WiFi ネットワークに接続します。
3. システムのデータリンクのプロパティを表示します。

```
# dladm show-linkprop link
```

4. リンクの固定された速度を設定します。



注意 - Oracle Solaris では、WiFi 接続の最適な速度が自動的に選択されます。初期のリンク速度を変更すると、パフォーマンスが低下したり、特定の WiFi 接続の確立が妨げられたりすることがあります。

次の例に示すように、リンク速度を、show-linkprop サブコマンドの出力にリスト表示されている指定可能な値のいずれかに変更できます。

```
# dladm set-linkprop -p speed=value link
```

5. リンク上のパケットフローを確認します。

```
# netstat -I net0 -i 5
input net0 output input (Total) output
packets errs packets errs colls packets errs packets errs colls
317 0 106 0 0 2905 0 571 0 0
14 0 0 0 0 20 0 0 0 0
7 0 0 0 0 16 0 1 0 0
5 0 0 0 0 9 0 0 0 0
304 0 10 0 0 631 0 316 0 0
338 0 9 0 0 722 0 381 0 0
294 0 7 0 0 670 0 371 0 0
306 0 5 0 0 649 0 338 0 0
289 0 5 0 0 597 0 301 0 0
```

例 7-2 リンクの速度の設定

次の例は、WiFi ネットワークに接続したあとにリンクの速度を設定する方法を示しています。

```
# dladm show-linkprop -p speed net0
```

```

LINK      PROPERTY      PERM VALUE      EFFECTIVE      DEFAULT      POSSIBLE
net0      speed          r- 25           0              0
1,2,5,6,9,11,12,18,24,36,48,54
# dladm set-linkprop -p speed=36 net0

# dladm show-linkprop -p speed net0
LINK      PROPERTY      PERM VALUE      EFFECTIVE      DEFAULT      POSSIBLE
net0      speed          r- 36           0              0
1,2,5,6,9,11,12,18,24,36,48,54

```

セキュアな WiFi 通信の確立

電波テクノロジーによって WiFi ネットワークが簡単に使用できるようになり、多くの場合ユーザーから自由にアクセスできます。その結果、WiFi ネットワークへの接続が安全でなくなる場合があります。

次のタイプの WiFi 接続は、よりセキュアです。

- アクセスが制限されたプライベート WiFi ネットワークへの接続。

企業や大学によって確立された内部ネットワークなどのプライベートネットワークは、ネットワークへのアクセスを、正しいセキュリティチャレンジを指定できるユーザーに制限しています。潜在的なユーザーは、接続シーケンス中に鍵を指定するか、またはセキュアな仮想プライベートネットワーク (VPN) アプリケーションを経由してネットワークにログインする必要があります。

- WiFi ネットワークへの接続の暗号化。

セキュリティ保護されたキーを使用して、システムと WiFi ネットワークの間の通信を暗号化できます。WiFi ネットワークへのアクセスポイントは、セキュアな鍵生成機能を備えた、自宅またはオフィスのルーターである必要があります。システムとルーターは、セキュリティ保護された接続を作成する前にキーを確立してから共有します。

`dladm` コマンドでは、アクセスポイントを経由した接続を暗号化するための WEP (Wired Equivalent Privacy) または WPA (Wi-Fi Protected Access) キーを使用できます。WEP プロトコルは、無線接続のための IEEE 802.11 仕様で定義されています。WPA プロトコルは、無線接続のための IEEE 802.11i 仕様で定義されています。Oracle Solaris は、WPA 標準のバージョン 1 および 2 をサポートしています。WEP および WPA に関連した `dladm` コマンドオプションの詳細は、[dladm\(1M\)](#) のマニュアルページを参照してください。

▼ WEP キーを指定して暗号化された WiFi ネットワーク接続を設定する方法

次の手順では、システムと自宅のルーターの間でセキュアな通信を設定する方法について説明します。家庭向けの無線および有線ルーターの多くには、セキュアな鍵を生成できる暗号化機能があります。

始める前に 自宅の無線ネットワークに接続する場合は、ルーターが構成され、WEP キーが生成されていることを確認してください。ルーターの製造元のドキュメントに従ってキー構成を生成し、保存します。

1. 管理者になります。
2. 次のようにして、WEP キーを含むセキュアオブジェクトを作成します。

```
# dladm create-secobj -c wep keyname
```

ここで、*keyname* はキーに付ける名前を表します。

3. セキュアオブジェクトに WEP キーの値を指定します。

`create-secobj` サブコマンドは次に、鍵の値を要求するスクリプトを実行します。

```
provide value for keyname: 5-or-13-byte key
confirm value for keyname: Retype key
```

この値は、ルーターによって生成されたキーです。スクリプトは、鍵の値として 5 バイトまたは 13 バイトの文字列 (ASCII または 16 進数の形式) のどちらかを受け入れます。

4. 今作成したキーの内容を表示します。

```
# dladm show-secobj
OBJECT          CLASS
keyname         wep
```

ここで、*keyname* は、このセキュアオブジェクトの名前です。

5. WiFi ネットワークへの暗号化された接続を作成します。

```
# dladm connect-wifi -e network -k keyname interface
```

6. 接続がセキュリティーで保護されていることを確認します。

```
# dladm show-wifi
LINK          STATUS          ESSID          SEC          STRENGTH      MODE          SPEED
```

```
net0      connected  wifi-1      wep      good      g      11Mb
```

前の出力では、SEC 列の下にある wep は、この接続に WEP 暗号化が適用されていることを示します。

例 7-3 WEP キーを使用して暗号化された WiFi 通信を設定する

次の例では、次のことがすでに実行されていることを前提にしています。

- ルーターの製造元のドキュメントに従って WEP キーを作成します。
- 鍵を保存することにより、それを使用してシステム上にセキュアオブジェクトを作成できるようにします。

次のようにして、セキュアオブジェクトを作成します。

```
# dladm create-secobj -c wep mykey
provide value for mykey: *****
confirm value for mkey: *****
```

ルーターによって生成された WEP キーを指定するときは、入力した値がアスタリスクでマスクされます。

次のコマンドは、セキュアオブジェクト mykey を使用して、WiFi ネットワーク citinet への暗号化された接続を確立します。

```
# dladm show-secobj
OBJECT          CLASS
mykey           wep
# dladm connect-wifi -e citinet -k mykey net0
```

次のコマンドは、WEP 暗号化を使用して citinet 無線ネットワークに接続されていることを検証します。

```
# dladm show-wifi
LINK      STATUS      ESSID      SEC      STRENGTH  MODE  SPEED
net0      connected  citinet    wep      good      g     36Mb
```

リアクティブモードでの既知の WLAN の管理

リアクティブプロファイルを使用している場合、システムが備える既知の WLAN への自動接続機能を活用することもできます。WLAN に接続する場合は、WLAN に関する情報が known-wlan タイプのネットワークプロファイルに保存されます。netcfg コマンドを使用して、known-wlan プロファイルを手動で作成することもできます。既知の WLAN の概要情報

は、[111 ページの「既知の WLAN の説明」](#)を参照してください。タスク関連の情報については、[133 ページの「既知の WLAN を作成する」](#)を参照してください。

デスクトップから無線ネットワークを管理する

デフォルトでは、無線ネットワーク接続が有効な場合、リアクティブネットワークデーモン (nwmnd) により、お気に入りリストで使用可能なネットワークへの接続が、リスト内の接続の優先度順に要求なしで試みられます。使用可能なお気に入りのネットワークがない場合は、「無線チューザ」ダイアログボックスが開きます。このダイアログボックスでは、参加する無線ネットワークを選択できます。

ネットワーク管理 GUI の「ネットワーク設定」ダイアログの「接続プロパティ」ビューの「無線」タブで、無線ネットワークへの接続が試行される方法を変更することもできます。必要に応じ、デスクトップにある「ネットワークステータス」通知アイコンを右クリックして、別の無線ネットワークに手動で接続することもできます。

ヒント - 「ネットワーク設定」ダイアログを使用して、選択したネットワークの「接続プロパティ」ビューにアクセスできます。このダイアログには、「表示」というラベルが付いたドロップダウンリストが付いています。このリストでは、特定のネットワークのビューを切り替えることができます。各ビューでは、実行可能なさまざまなタスクや、選択したネットワークに関する、該当ビューに固有の情報が提供されます。

次のビューは、システム上にある各ネットワークプロファイルのネットワーク接続ごとに存在します。

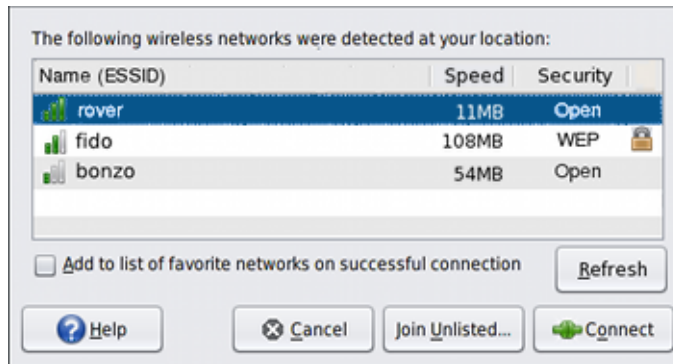
- 接続ステータス
- ネットワークプロファイル
- 接続プロパティ

プロファイルベースのネットワーク構成の詳細は、[第5章「Oracle Solaris でのプロファイルベースのネットワーク構成の管理について」](#)および GUI オンラインヘルプを参照してください。

▼ 無線ネットワークに加入する方法

無線ネットワークに加入するには、「ネットワークステータス」通知アイコンを右クリックすると使用可能になる「無線ネットワークの加入」オプションを選択します。「無線チューザ」ダイアログには、接続可能な無線ネットワークのリストが表示されます。

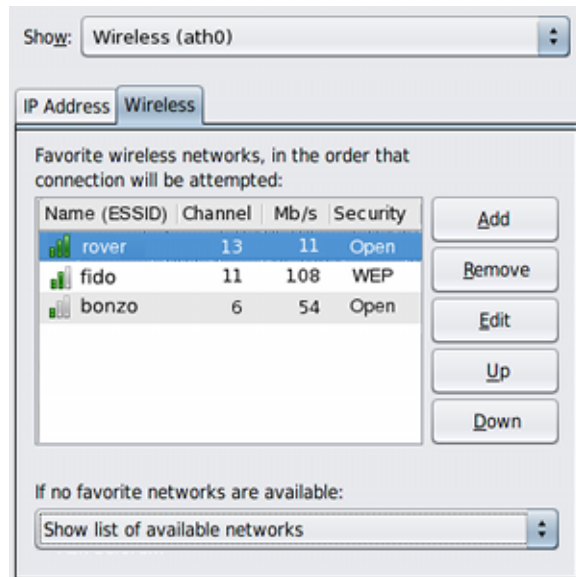
1. 別の無線ネットワークに手動で接続するには、次のいずれかを実行します。
 - 「ネットワークステータス」通知アイコンの右クリックメニューから、使用可能な無線ネットワークを選択します。
 - 「ネットワークステータス」通知のアイコンメニューから「リストにない無線ネットワークに加入」オプションを選択します。
リストにない無線ネットワークとは、そのネットワーク名をブロードキャストしないように構成されているけれども、引き続き加入が可能な無線ネットワークのことです。
 - 「無線チューザ」ダイアログから、使用可能な無線ネットワークを選択します。加入可能な無線ネットワークの選択肢がある場合は、このダイアログが自動的に表示されます。



2. 「無線ネットワークの加入」ダイアログが開いたら、選択した無線ネットワークに関するすべての必要情報を指定してください。

デスクトップからお気に入りの無線ネットワークを管理する

デフォルトでは、はじめて無線ネットワークに参加すると、「無線ネットワークに加入」ダイアログに「接続が成功したらお気に入りのネットワークのリストに追加する」という名前のチェックボックスが表示されます。



お気に入りリストに無線ネットワークを追加するには (接続に成功したら)、このボックスを選択します。お気に入りリストにネットワークを追加しない場合は、ボックスの選択を解除します。ボックスはデフォルトで選択されています。

現在使用できない、または現在お気に入りリストにネットワーク名をブロードキャストしていない無線ネットワークを追加するには、「接続プロパティ」ビューの「無線」タブに移動してから、「追加」ボタンをクリックします。ネットワークを追加するには、ネットワーク名、セキュリティータイプ、およびセキュリティーキーを知っている必要があります。

索引

数字・記号

- /etc/hosts ファイル, 52
- /etc/inet/ndpd.conf ファイル
 - 一時アドレスの構成, 60

あ

- 新しい機能
 - IPv6 の一時アドレス, 59
 - リンクローカルアドレスの手動構成, 62
- アドレス
 - 一時、IPv6 の, 59
- アドレスオブジェクト, 52
- アドレス解決プロトコル (ARP), 50
- 一時アドレス、IPv6 の
 - 構成, 59
 - 定義, 59
- インタフェース
 - 構成
 - 一時アドレス, 59
 - 手動、IPv6 用, 56
- インタフェース ID
 - 手動構成したトークンの使用, 64

か

- 仮想ネットワークカード (VNIC), 32
- 仮想プライベートネットワーク (VPN), 39
- 仮想ローカルエリアネットワーク (VLAN), 32
- 既知の WLAN, 133
- 既知の WLAN プロファイル
 - 対話形式で作成する, 133
- 構成
 - インタフェースを手動で、IPv6 用, 56

さ

- サーバー、IPv6
 - IPv6 を有効にする, 64
- 最上位のプロファイル, 122
- 自動ネゴシエーション, 36
- ジャンボフレーム
 - サポートの有効化, 37
- 自律システム (AS) 参照 ネットワークトポロジ
- 静的ルーティング
 - 構成例, 70
 - 最適な使用対象, 67
 - 静的ルートの追加, 69
 - ホストでの手動構成, 71
- セキュリティと承認, 115
- セキュリティの注意点
 - WiFi, 157
- 全二重, 38

た

- 対称ルーティング, 78
- 対話形式で作成する
 - ENM プロファイル, 132
 - NCU を含む NCP, 127
 - 既知の WLAN プロファイル, 133
- 単一のプロパティ値を対話形式で取得する, 140
- データリンク
 - autopush プロパティ, 39
 - Ethernet パラメータ値, 39
 - MTU サイズの変更, 37
 - STREAMS モジュール, 39
 - VLAN, 32
 - VNIC, 32
 - カスタマイズされた名前の使用規則, 30
 - 削除, 34
 - 自動ネゴシエーション, 36

通知速度と有効化速度, 37
名前変更, 35
汎用名, 35
表示
 一般情報, 32
 システム上の物理的场所, 34
 ネットワークドライバのプロパティ, 39, 40
 物理属性, 33
 リンク, 32
 リンクプロパティ, 39
物理リンク, 32
プロパティの設定, 36
ランタイム統計の取得, 35
リンクアグリゲーション, 32
リンク速度, 37
リンク名, 29
動的再構成 (DR)
 NIC の交換, 43
動的ルーティング
 最適な使用対象, 67
特定のプロパティの値を取得する, 140
ドメイン名
 nis/domain SMF サービス, 96

な

ネットワークインタフェースカード (NIC)
 交換, DR を使用, 43
ネットワーク構成コマンド, 18
ネットワーク構成ツール
 dladm コマンド, 31
ネットワーク構成の承認
 とセキュリティ, 115
ネットワークポロジ
 自律システム, 72
ネットワークの構成
 IPv6 対応のマルチホームホスト, 56
ネットワークプロファイル
 ENM, 107
 NCU, 107
 エクスポートされたプロファイルを復元する, 148
 既知の WLAN, 107
 作成する
 場所プロファイル, 129
 システム上のプロファイル情報をリスト表示する,
 137

場所プロファイル, 107
プロファイル構成をエクスポートおよび復元する,
146
プロファイルの作成と管理, 105
プロファイルのプロパティ値を設定および変更す
る, 135
プロファイルを削除する, 145
ネットワークプロファイル構成
 作成する
 場所プロファイル, 129
 システム上のプロファイル情報をリスト表示する,
 137
 プロファイル構成をエクスポートおよび復元する,
 146
 プロファイルのプロパティ値を設定および変更す
 る, 135
ネットワークプロファイルを作成する
 ENM を作成する, 131

は

パケット転送
 インタフェース上の, 80
場所プロファイル
 対話形式で作成する, 130
 場所プロファイルを作成する, 129
半二重, 38
表示する
 補助的な状態の値, 144
プライマリインタフェース, 切り替え, 35, 41, 83
プロパティ
 activation-mode プロパティ, 114
 walkprop サブコマンドを使用してプロパティ値を
 変更する, 141
 単一のプロパティ値を対話形式で取得してリスト
 表示する, 140
 特定のプロパティの値を取得する, 140
 特定のプロファイルのすべてのプロパティ値をリ
 スト表示する, 138
 プロパティ値を対話形式で設定する, 136
 プロファイルのプロパティ値を設定および変更す
 る, 135
 プロパティ値を対話形式で設定する, 136
プロファイル
 ネットワークプロファイルタイプ, 106
 プロファイル構成をエクスポートおよび復元する, 146

- プロファイルのアクティブ化ポリシー, 113
 - プロファイルのプロパティ値を設定および変更する, 135
 - コマンド行モードでプロパティ値を設定する, 137
 - プロパティ値を対話形式で設定する, 137
 - プロファイルを削除する, 145
 - プロファイルを作成する
 - ENM を作成するプロファイル
 - ENM プロファイル, 131
 - ENM を対話形式で作成する, 132
 - 作成する
 - NCP に NCU を作成する, 126
 - WLAN を作成する, 133
 - 場所プロファイル, 129
 - 対話形式で作成する
 - NCU を含む NCP, 127
 - 既知の WLAN プロファイル, 133
 - プロファイルを復元する, 148
 - プロファイルを無効にする, 121
 - プロファイルを有効にする, 121
 - 変更
 - プロファイルのプロパティ値, 135
 - 変更する
 - walkprop サブコマンドを使用してプロパティ値を変更する, 141
 - 補助的な状態の値
 - 表示する, 144
 - ホスト
 - IPv6 一時アドレス, 59
 - マルチホーム
 - 構成, 75
- ま**
- マルチホームホスト, 78
 - IPv6 用に有効化, 56
 - 定義, 75
 - 無線インタフェース, 151
- ら**
- ランタイム統計
 - データリンク
 - dlstat, 35
 - リアクティブネットワーク構成
 - プロファイル構成をエクスポートする, 148
 - プロファイルを削除する, 145
 - リアクティブネットワーク構成モード
 - ネットワークプロファイルとタイプ, 106
 - リアクティブプロファイル
 - プロファイルのアクティブ化ポリシー, 113
 - リアクティブプロファイルの作成と管理, 105
 - リスト表示
 - すべてのプロファイル, 137
 - 特定のプロファイルのすべてのプロパティ値, 138
 - リスト表示する
 - システム上のプロファイル情報, 137
 - 特定のプロパティの値, 140
 - リモートアドレス, 52
 - リンクアグリゲーション, 32
 - リンク速度, 37
 - リンクのローカルアドレス
 - 手動構成、トークン, 64
 - リンク名, 35
 - 汎用, 27
 - ルーティング
 - IPv6, 74
 - 静的構成, 71
 - 単一インタフェースホストの, 71
 - ルーティングテーブル
 - 手動構成, 69
 - 例
 - プロパティ値を対話形式で設定する, 137
 - プロファイル構成をエクスポートする, 147
 - コマンドファイルモードで, 148
 - プロファイルを無効にする, 121
 - プロファイルを有効にする, 121
 - ローカルアドレス, 52
- A**
- activation-mode プロパティ, 114
 - さまざまなプロファイルタイプのプロパティ値, 114
 - autopush プロパティ, 39
- B**
- BSSID, 152

C

cfgadm コマンド, 44
CIDR 表記法, 52

D

DefaultFixed の場所, 135
DHCP, 53
dladm コマンド, 18, 31
 delete-phys, 34
 rename-link, 35
 reset-linkprop, 36
 scan-wifi, 152
 set-linkprop, 36
 show-ether, 39, 40
 show-link, 32
 show-linkprop, 39, 156
 show-phys, 33
 show-wifi, 153
dlstat コマンド, 35

E

ECMP, 78
ENM
 ENM プロファイルを対話形式で作成する, 132
ESSID, 152
Ethernet パラメータ, 39

I

ICMP, 50
ifconfig コマンド, 50
IP アドレス
 DHCP, 53
 IPv4 と IPv6, 52
 削除, 84
 静的, 52
 パケット転送, 80
 プロパティ, 81, 89
 モニタリング, 85
 ローカルおよびリモート, 52
IP インタフェース
 IP アドレス, 88, 89
 IP アドレスの削除, 84

 IP アドレスの変更, 84
 IP アドレスの割り当て, 52
 MAC アドレスが一意であることの確認, 45
 アドレスプロパティ, 81
 インタフェース構成の削除, 83
 インタフェースのプロパティ, 87
 構成, 54
 パケット転送の有効化, 80
 表示
 IP アドレス, 88
 アドレスのプロパティ, 89
 一般情報, 53, 85
 インタフェース, 86
 インタフェースのプロパティ, 87
 プライマリインタフェースの変更, 83
 無効化と有効化, 83
 モニタリング, 85, 86

IP トンネル, 52

 ローカルアドレスとリモートアドレス, 52

ipadm コマンド, 18

 create-addr, 52
 create-ip, 51
 delete-addr, 84
 delete-ip, 83
 disable-if, 83
 set-addrprop, 81
 show-addr, 88
 show-addrprop, 81, 89
 show-if, 86
 show-ifprop, 87
 マルチホームホスト, 76

IPv6

 一時アドレスの構成, 59
 有効にする、サーバー上で, 64
 ルーティング, 74

M

MAC アドレス
 一意であることの確認, 45
MTU, 37

N

name-service/switch サービス, 52

- NCP
 - NCU を含む NCP を作成する
 - NCU を作成する, 127
 - NCU
 - 作成する, 126
 - ndpd.conf ファイル
 - 一時アドレスの構成, 60
 - netadm コマンド, 18
 - 対話型モードnetcfg コマンド, 121
 - netcfg コマンド, 18
 - walkprop サブコマンド, 141
 - 対話型モード, 121
 - netstat コマンド
 - WiFi リンク上のパケットフローの確認, 156
 - nis/tdomain SMF サービス
 - ローカルファイルモード構成, 96
 - 暗号化された通信の例, 159
 - 拡張サービスセット識別子 (ESSID), 152
 - 基本サービスセット識別子 (BSSID), 152
 - セキュアな WiFi リンク, 157
 - 接続の暗号化, 158
 - 定義, 151
 - パケットフローの確認, 156
 - リンクのモニタリング, 156
 - 例、リンク速度の設定, 156
- P**
- Power Management, 38
- R**
- route コマンド, 18, 54
- S**
- SCTP, 50
 - STREAMS モジュール
 - とデータリンク, 39
- U**
- UDP, 50
- W**
- walkprop サブコマンド
 - プロパティ値を表示および変更する, 141
 - WiFi
 - IEEE 802.11 仕様, 151
 - WiFi ネットワークへの接続, 151
 - WiFi の構成例, 155

