

# Oracle® Solaris 11.1에서 고정된 네트워크 구성을 사용하여 시스템 연결

Copyright © 2011, 2012, Oracle and/or its affiliates. All rights reserved.

본 소프트웨어와 관련 문서는 사용 제한 및 기밀 유지 규정을 포함하는 라이선스 계약서에 의거해 제공되며, 지적 재산법에 의해 보호됩니다. 라이선스 계약서 상에 명시적으로 허용되어 있는 경우나 법규에 의해 허용된 경우를 제외하고, 어떠한 부분도 복사, 재생, 번역, 방송, 수정, 라이선스, 전송, 배포, 진열, 실행, 발행, 또는 전시될 수 없습니다. 본 소프트웨어를 리버스 엔지니어링, 디스어셈블리 또는 디컴파일하는 것은 상호 운용에 대한 법규에 의해 명시된 경우를 제외하고는 금지되어 있습니다.

이 안의 내용은 사전 공지 없이 변경될 수 있으며 오류가 존재하지 않음을 보증하지 않습니다. 만일 오류를 발견하면 서면으로 통지해 주시기 바랍니다.

만일 본 소프트웨어나 관련 문서를 미국 정부나 또는 미국 정부를 대신하여 라이선스한 개인이나 법인에게 배송하는 경우, 다음 공지 사항이 적용됩니다.

#### 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.

본 소프트웨어 혹은 하드웨어는 다양한 정보 관리 애플리케이션의 일반적인 사용을 목적으로 개발되었습니다. 본 소프트웨어 혹은 하드웨어는 개인적인 상해를 초래할 수 있는 애플리케이션을 포함한 본질적으로 위험한 애플리케이션에서 사용할 목적으로 개발되거나 그 용도로 사용될 수 없습니다. 만일 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서 사용할 경우, 라이선스 사용자는 해당 애플리케이션의 안전한 사용을 위해 모든 적절한 비상-안전, 백업, 대비 및 기타 조치를 반드시 취해야 합니다. Oracle Corporation과 그 회사는 본 소프트웨어 혹은 하드웨어를 위험한 애플리케이션에서의 사용으로 인해 발생하는 어떠한 손해에 대해서도 책임지지 않습니다.

Oracle과 Java는 Oracle Corporation 및/또는 그 자회사의 등록 상표입니다. 기타의 명칭들은 각 해당 명칭을 소유한 회사의 상표일 수 있습니다.

Intel 및 Intel Xeon은 Intel Corporation의 상표 내지는 등록 상표입니다. SPARC 상표 일체는 라이선스에 의거하여 사용되며 SPARC International, Inc.의 상표 내지는 등록 상표입니다. AMD, Opteron, AMD 로고, 및 AMD Opteron 로고는 Advanced Micro Devices의 상표 내지는 등록 상표입니다. UNIX는 The Open Group의 등록 상표입니다.

본 소프트웨어 혹은 하드웨어와 관련 문서(설명서)는 제 3자로부터 제공되는 콘텐츠, 제품 및 서비스에 접속할 수 있거나 정보를 제공합니다. Oracle Corporation과 그 자회사는 제 3자의 콘텐츠, 제품 및 서비스와 관련하여 어떠한 책임도 지지 않으며 명시적으로 모든 보증에 대해서도 책임을 지지 않습니다. Oracle Corporation과 그 자회사는 제 3자의 콘텐츠, 제품 및 서비스에 접속하거나 사용으로 인해 초래되는 어떠한 손실, 비용 또는 손해에 대해 어떠한 책임도 지지 않습니다.

# 목차

---

머리말 .....	7
<b>1 고정된 네트워크 구성 개요 .....</b>	<b>9</b>
고정된 네트워크 구성이란? .....	9
프로파일로 관리되는 네트워크 구성의 주요 내용 .....	10
네트워크 구성 도구 .....	11
dladm 명령 .....	12
ipadm 명령 .....	13
<b>2 네트워크에 대한 시스템 구성 .....</b>	<b>15</b>
네트워크 구성(작업 맵) .....	15
▼ SPARC: 각 인터페이스의 MAC 주소가 고유한지 확인하는 방법 .....	16
▼ 시스템의 활성 NCP를 변경하는 방법 .....	17
▼ IP 인터페이스를 구성하는 방법 .....	18
기타 네트워크 구성 및 관리 작업 .....	22
<b>3 데이터 링크 작업 .....</b>	<b>25</b>
기본 dladm 명령 .....	25
데이터 링크에 대한 일반 정보 표시(dladm) .....	25
시스템 데이터 링크 표시(dladm show-link) .....	26
데이터 링크의 물리적 속성 표시(dladm show-phys) .....	26
데이터 링크 삭제(dladm delete-phys) .....	27
데이터 링크 이름 바꾸기(dladm rename-link) .....	27
데이터 링크 등록 정보 사용자 정의 .....	28
데이터 링크 등록 정보 개요 .....	28
점보 프레임 지원 사용 .....	28
링크 속도 매개변수 수정 .....	29

데이터 링크에 STREAMS 모듈 설정 .....	30
직접 메모리 액세스 바인딩을 사용하도록 e1000g 드라이버 설정 .....	31
인터럽트 속도를 수동으로 설정 .....	31
데이터 링크 등록 정보에 대한 상태 정보 가져오기 .....	32
dladm 명령을 사용한 기타 구성 작업 .....	33
▼ 시스템에서 주 인터페이스를 전환하는 방법 .....	34
▼ 동적 재구성을 사용하여 네트워크 인터페이스 카드를 교체하는 방법 .....	35
<b>4 IP 인터페이스 작업 .....</b>	<b>39</b>
기본 ipadm 명령 .....	39
IP 인터페이스 구성 제거(ipadm delete-ip) .....	39
IP 인터페이스 구성 사용 안함(ipadm disable-ip) .....	40
인터페이스의 주소 제거(ipadm delete-addr) .....	40
IP 인터페이스 등록 정보 설정 .....	41
패킷 전달 사용 .....	41
IP 주소 등록 정보 설정 .....	42
TCP/IP 프로토콜 등록 정보 설정 .....	43
전역적으로 패킷 전달 사용 .....	44
권한이 부여된 포트 설정 .....	45
멀티홈 호스트에 대칭 경로 지정 구현 .....	46
트래픽 혼잡 제어 구현 .....	47
TCP 수신 버퍼 크기 변경 .....	48
IP 인터페이스 및 주소 모니터링 .....	50
IP 인터페이스에 대한 일반 정보 가져오기 .....	50
IP 인터페이스에 대한 정보 가져오기 .....	51
IP 인터페이스 등록 정보에 대한 정보 가져오기 .....	52
IP 주소에 대한 정보 가져오기 .....	53
IP 주소 등록 정보에 대한 정보 가져오기 .....	54
<b>5 Oracle Solaris 실행 랩탑에서 무선 네트워킹 구성 .....</b>	<b>55</b>
WiFi 통신 작업 맵 .....	55
▼ WiFi 네트워크에 연결하는 방법 .....	56
▼ WiFi 링크를 모니터링하는 방법 .....	59
보안 WiFi 통신 .....	61
▼ 암호화된 WiFi 네트워크 연결을 설정하는 방법 .....	61

**A** 비교 맵:ifconfig 및 ipadm 명령 ..... 65

**B** 비교 맵:ndd 및 ipadm 명령 ..... 69

색인 .....73



# 머리말

---

**Oracle Solaris 11.1**에서 고정된 네트워크 구성을 사용하여 시스템 연결을 시작합니다. 이 설명서는 Oracle Solaris 네트워크 구성을 위한 기본 항목 및 절차가 포함된 **Oracle Solaris 11.1 네트워크 설정** 시리즈의 일부입니다. 이 설명서에서는 Oracle Solaris를 이미 설치했다고 가정합니다. 따라서 네트워크를 구성하거나 네트워크에 필요한 네트워킹 소프트웨어를 구성할 준비가 되어 있어야 합니다.

---

주 - 본 Oracle Solaris 릴리스는 프로세서 아키텍처의 SPARC 및 x86 제품군을 사용하는 시스템을 지원합니다. 지원되는 시스템은 **Oracle Solaris OS: Hardware Compatibility Lists**를 참조하십시오. 이 설명서에서는 플랫폼 유형에 따른 구현 차이가 있는 경우 이에 대하여 설명합니다.

지원되는 시스템은 **Oracle Solaris OS: Hardware Compatibility Lists**를 참조하십시오.

---

## 이 설명서의 대상

이 설명서는 네트워크에 구성된 Oracle Solaris 실행 시스템의 관리 책임자를 대상으로 작성되었습니다. 이 설명서를 사용하려면 2년 이상의 UNIX 시스템 관리 경험이 있어야 합니다. UNIX 시스템 관리 교육 과정에 참석하는 것도 도움이 될 수 있습니다.

## Oracle Support에 액세스

Oracle 고객은 My Oracle Support를 통해 온라인 지원에 액세스할 수 있습니다. 자세한 내용은 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>를 참조하거나, 청각 장애가 있는 경우 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>를 방문하십시오.

## 활자체 규약

다음 표는 이 설명서에서 사용되는 활자체 규약에 대해 설명합니다.

표 P-1 활자체 규약

활자체 또는 기호	설명	예제
AaBbCc123	명령, 파일, 디렉토리 이름 및 컴퓨터 화면에 출력되는 내용입니다.	.login 파일을 편집하십시오.  모든 파일 목록을 보려면 <code>ls -a</code> 명령을 사용하십시오.  machine_name% you have mail.
AaBbCc123	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	machine_name% <b>su</b>  Password:
AaBbCc123	위치 표시자: 실제 이름이나 값으로 바뀝니다.	<code>rm filename</code> 명령을 사용하여 파일을 제거합니다.
AaBbCc123	설명서 제목, 새 용어, 강조 표시할 용어입니다.	<b>사용자 설명서</b> 의 6장을 읽으십시오.  캐시는 로컬로 저장된 복사본입니다.  파일을 저장하면 <b>안 됩니다</b> .  <b>주:</b> 일부 강조된 항목은 온라인에서 굵은체로 나타납니다.

## 명령 예의 셸 프롬프트

다음 표에는 Oracle Solaris OS에 포함된 셸의 기본 UNIX 시스템 프롬프트 및 슈퍼유저 프롬프트가 나와 있습니다. 명령 예제에 표시된 기본 시스템 프롬프트는 Oracle Solaris 릴리스에 따라 다릅니다.

표 P-2 셸 프롬프트

셸	프롬프트
Bash 셸, Korn 셸 및 Bourne 셸	\$
슈퍼유저용 Bash 셸, Korn 셸 및 Bourne 셸	#
C 셸	machine_name%
슈퍼유저용 C 셸	machine_name#

## 고정된 네트워크 구성 개요

---

Oracle Solaris 11을 실행하는 시스템에서 네트워크 구성은 시스템에서 활성 상태인 NCP(네트워크 구성 프로파일)에 따라 반응적이거나 고정적일 수 있습니다. 반응적 및 고정된 네트워크 구성에 대한 개요는 **Oracle Solaris 11 네트워크 소개**를 참조하십시오. NCP를 만들고 구성하는 방법에 대한 자세한 내용은 **Oracle Solaris 11.1에서 반응적 네트워크 구성을 사용하여 시스템 연결**을 참조하십시오.

이 장에서는 고정된 네트워크 구성에 대한 일반적인 소개를 제공하고 다음과 같은 항목에 대해 다룹니다.

- 9 페이지 “고정된 네트워크 구성이란?”
- 10 페이지 “프로파일로 관리되는 네트워크 구성의 주요 내용”
- 11 페이지 “네트워크 구성 도구”

### 고정된 네트워크 구성이란?

Oracle Solaris 11에서 네트워크 구성은 NCP(네트워크 구성 프로파일)로 관리됩니다. 특정 시스템에서 작동하는 NCP 유형에 따라 시스템의 네트워크 구성이 결정됩니다. NCP가 반응적인 경우 해당 시스템의 네트워크 구성이 동적으로 구현됩니다. NCP가 고정된 경우 네트워크 구성이 정적으로 구현됩니다.

고정된 네트워크 구성은 특정 네트워크 설정이 시스템에서 인스턴스화되는 구성 모드를 나타냅니다. 반응적 네트워크 구성 모드와 달리, 고정 구성 모드에서 인스턴스화된 구성은 시스템의 네트워크 환경에 관계없이 변경되지 않은 상태로 유지됩니다. 해당 환경에서 인터페이스 추가 등의 변경 사항이 발생할 경우, 시스템이 새 환경에 맞게 적용되도록 시스템의 네트워크 설정을 수동으로 재구성해야 합니다.

주 - 고정된 네트워크 구성을 단순한 정적 IP 주소 구성과 혼동하지 마십시오. 고정된 네트워크 구성에서는 인터페이스에 DHCP 주소를 지정할 수 있습니다. 마찬가지로, 반응적 네트워크 구성에서는 인터페이스가 정적 IP 주소로 구성되는 NCP를 만들 수 있습니다. 따라서 고정된 네트워크 구성은 범위가 보다 포괄적이며, 시스템 환경의 변경 사항에 따라 시스템의 네트워크 구성을 변경할 수 있는 기능을 나타냅니다.

다음 표에서는 네트워크 구성의 두 가지 모드를 서로 비교하여 보여줍니다.

기능	반응적 네트워크 구성	고정된 네트워크 구성
시스템의 네트워크 환경 변경 사항에 자동으로 대응	구성 가능한 여러 NCP를 통해 지원됨	지원 안됨, 필요에 따라 수동 재구성 필요
시스템에서 작동하는 NCP 유형	반응적(Automatic 또는 다른 사용자가 만든 일부 NCP)	고정된(DefaultFixed)
다중 NCP	지원됨(하지만 NCP는 한 번에 하나만 활성화 상태일 수 있음)	지원되지 않음
사용자가 만든 NCP	지원됨	시스템에서 생성된 고정된 NCP가 하나만 존재합니다(DefaultFixed). 하지만 DefaultFixed 내용은 완전히 사용자에게 의해 결정됩니다.

다음 절에서는 프로파일로 관리되는 네트워크 구성에 대해 보다 자세히 설명하고 네트워크 구성을 위해 사용되는 도구에 대해 설명합니다.

## 프로파일로 관리되는 네트워크 구성의 주요 내용

Oracle Solaris 11에서 네트워크 구성은 프로파일을 기반으로 합니다. 시스템의 네트워크 구성은 NCP 및 해당 위치 프로파일에 의해 관리됩니다. 프로파일로 관리되는 네트워크 구성에 대한 소개 정보를 보려면 [Oracle Solaris 11 네트워킹 소개](#)의 “네트워크 구성 프로파일”을 참조하십시오. NCP에 대한 자세한 내용은 [Oracle Solaris 11.1에서 반응적 네트워크 구성을 사용하여 시스템 연결](#)을 참조하십시오.

주 - 네트워크 구성에 대해 주요 프로파일 유형은 NCP, 위치 프로파일, ENM(외부 네트워크 수정자) 및 WLAN(무선 LAN)입니다. 이러한 유형 중 기본 프로파일은 NCP입니다. 달리 지정되지 않은 경우 이 설명서 전체에서 **프로파일**이란 용어는 NCP를 가리킵니다.

프로파일 기반 네트워크 구성의 주요 내용은 다음과 같습니다.

- 한 번에 한 쌍의 NCP 및 위치 프로파일만 활성화되어 시스템의 네트워크 구성을 관리할 수 있습니다. 시스템의 다른 기존 NCP는 모두 작동하지 않습니다.
- 활성화 NCP는 **반응적** 또는 **고정된** 프로파일일 수 있습니다. 반응적 프로파일을 사용할 경우 네트워크 구성을 모니터링하여 시스템 네트워크 환경의 변경 사항에 맞게 조정합니다. 고정된 프로파일을 사용할 경우 네트워크 구성이 인스턴스화되지만 모니터링되지는 않습니다.
- 활성화 NCP가 반응적인 경우 시스템의 네트워크 구성이 적응적입니다. 활성화 NCP가 고정된 경우 시스템의 네트워크 구성이 항상적입니다.
- NCP의 여러 등록 정보 값으로 구성된 정책이 프로파일에서 네트워크 구성을 관리하는 방식을 제어합니다.
- NCP 등록 정보에 대한 변경 사항은 새 등록 정보 값으로 즉시 구현됩니다. 이러한 새 값은 네트워크 구성을 관리하는 프로파일의 정책에 속합니다.

시스템이 고정된 네트워킹에 맞게 구성된 경우 네트워크 구성을 관리하는 활성화 NCP는 `DefaultFixed`입니다. 이 프로파일은 OS에서 생성되며 시스템에서 유일한 고정된 프로파일입니다. 시스템에서는 여러 개의 고정된 프로파일이 지원되지 않습니다.

`DefaultFixed` NCP의 등록 정보는 `DefaultFixed` NCP가 활성화 상태일 때 생성되었거나 고정된 지속 구성을 반영합니다.

## 네트워크 구성 도구

Oracle Solaris 11에서는 네트워크 구성을 위해 네 가지 네트워크 명령이 제공됩니다.

- `netcfg` 명령
- `netadm` 명령
- `dladm` 명령
- `ipadm` 명령

`netcfg` 및 `netadm` 명령은 시스템에서 반응적 네트워크 구성을 관리하는 데 사용됩니다. `netcfg` 명령을 사용하여 반응적 네트워크 구성을 구현하는 프로파일(NCP, 위치 프로파일, ENM 및 WLAN)을 만들고 구성합니다. 하지만 고정된 네트워크 구성의 시스템에서는 `netcfg` 명령을 사용하여 `DefaultFixed` 프로파일만 볼 수 있습니다. `netadm` 명령은 시스템에서 모든 프로파일을 관리하는 데 사용되며, 특히 시스템의 네트워크 프로파일을 나열하고 활성화 NCP를 다른 항목으로 바꾸는 등의 작업을 위해 사용됩니다.

`dladm` 및 `ipadm` 명령은 각각 데이터 링크 및 IP 인터페이스를 구성하는 데 사용됩니다. 이러한 명령은 지속 구성을 만들고, 명령을 사용할 때 시스템에서 활성화 상태인 프로파일에 적용됩니다.

예를 들어, 데이터 링크 `net0`이 MTU(최대 전송 단위) 1200으로 구성되었고 활성화 NCP가 `Automatic`이면 이 MTU 값이 `Automatic` NCP에서 `net0`에 대해 지속적으로 적용됩니다. 그런 후 `myncp`라는 두 번째 NCP를 활성화했다고 가정해보십시오. `dladm` 명령을 실행하여 MTU를 다른 값으로 설정하면 이 값이 `myncp`에 적용됩니다. 따라서 `net0`은 다른

프로파일에서 서로 다른 MTU 값을 가질 수 있습니다. 따라서 `dladm` 및 `ipadm` 명령을 사용해서도 간접적으로 프로파일을 구성할 수 있습니다.

이러한 `dladm` 또는 `ipadm` 명령을 사용하여 데이터 링크 및 IP 인터페이스를 구성할 때는 다음과 같은 사용 범위에 주의하십시오.

- 이 두 명령은 활성 프로파일의 데이터 링크 및 IP 인터페이스만 구성합니다. 기본 경로 설정과 같은 프로파일의 다른 등록 정보를 구성하려면 `netcfg` 명령을 사용하여 기본 경로를 참조하는 프로파일의 등록 정보를 구성합니다. 또는 `routeadm` 명령을 사용하여 시스템의 경로 지정 테이블에서 기본 경로를 직접적으로 설정합니다. 후자의 경우 구성이 시스템에서 활성 상태인 프로파일에 적용됩니다.
- 프로파일이 활성 상태이면 어떠한 반응적 프로파일에서도 `dladm` 및 `ipadm` 명령을 사용할 수 있습니다. 하지만 시스템의 유일한 고정된 프로파일인 `DefaultFixed` 프로파일은 `netcfg` 명령을 사용하여 구성할 수 없습니다. `netadm` 및 `netcfg` 명령을 사용하여 `DefaultFixed` 프로파일의 등록 정보를 볼 수만 있고 이를 구성할 수는 없습니다.

`dladm` 및 `ipadm` 명령은 반응적 프로파일이든 고정된 프로파일이든 간에 활성 프로파일에 적용됩니다. 따라서 두 명령을 사용하기 전에 다음을 확인해야 합니다.

- 올바른 대상 프로파일을 변경하도록 보장하기 위해서는 시스템에서 활성 상태인 프로파일을 확인해야 합니다.
- 명령을 사용한 후 예기치 않은 구성 동작이 발생하지 않도록 대상 프로파일이 반응적 프로파일인지 또는 고정된 프로파일인지 확인합니다. 반응적 프로파일은 고정된 프로파일과 다른 방식으로 네트워크 구성을 관리합니다. 따라서 변경 사항을 구현할 때도 두 프로파일의 동작이 서로 다릅니다.

다음 절에서는 `dladm` 및 `ipadm` 명령에 대해 자세히 설명합니다.

## dladm 명령

`dladm` 명령을 사용하여 데이터 링크를 구성합니다. 링크의 네트워크 드라이버가 `e1000g`와 같은 GLDv3 드라이버 구성 프레임워크로 변환된 경우 `dladm` 명령을 사용하여 데이터 링크 등록 정보를 사용자 정의할 수 있습니다. 특정 드라이버가 이 기능을 지원하는지 여부를 확인하려면 드라이버의 매뉴얼 페이지를 참조하십시오.

GLDv3 드라이버 구성 프레임워크의 전체 구현으로 다음과 같이 NIC(네트워크 인터페이스 카드) 드라이버의 구성이 향상되었습니다.

- 네트워크 드라이버 등록 정보를 구성하려면 단일 명령 인터페이스인 `dladm` 명령만 있으면 됩니다.
- 등록 정보에 관계없이 동일한 구문이 사용됩니다. `dladm subcommand properties datalink`.
- `dladm` 명령 사용은 드라이버의 공용 및 개인 등록 정보에 모두 적용됩니다.

- 특정 드라이버에서 `dladm` 명령을 사용하는 경우 비슷한 유형의 다른 NIC 네트워크 연결은 중단되지 않습니다. 따라서 데이터 링크 등록 정보를 동적으로 구성할 수 있습니다.
- 데이터 링크 구성 값은 `dladm` 저장소에 저장되며 시스템을 재부트한 후에도 지속됩니다.

데이터 링크를 구성할 때 이러한 장점을 활용하려면 `ndd` 명령과 같은 이전 릴리스의 사용자 정의 도구 대신 `dladm`을 구성 도구로 사용해야 합니다.

`dladm` 명령에 대한 자세한 내용은 `dladm(1M)` 매뉴얼 페이지를 참조하십시오. `dladm` 명령에 사용할 수 있는 하위 명령 목록을 보려면 다음을 입력합니다.

```
# dladm help
The following subcommands are supported:
Bridge      : add-bridge      create-bridge    delete-bridge
              modify-bridge  remove-bridge    show-bridge
Etherstub   : create-etherstub delete-etherstub show-etherstub
IB          : create-part    delete-part      show-ib          show-part
IP tunnel   : create-iptun   delete-iptun     modify-iptun     show-iptun
Link Aggregation: add-aggr      create-aggr      delete-aggr
              modify-aggr    remove-aggr      show-aggr
Link        : rename-link    reset-linkprop   set-linkprop
              show-link      show-linkprop
Secure Object : create-secobj  delete-secobj    show-secobj
VLAN        : create-vlan   delete-vlan      modify-vlan      show-vlan
VNIC        : create-vnic   delete-vnic      modify-vnic      show-vnic
Wifi        : connect-wifi  disconnect-wifi  scan-wifi        show-wifi
Miscellaneous : delete-phys  show-ether       show-phys        show-usage
For more info, run: dladm help <subcommand>.
```

데이터 링크에 대해 `dladm` 명령을 사용하려면 3 장, “데이터 링크 작업”을 참조하십시오.

## ipadm 명령

Oracle Solaris의 향상 기능이 기존 도구의 기능보다 훨씬 효율적으로 다양한 네트워크 구성 측면을 관리합니다. 예를 들어, `ifconfig` 명령은 네트워크 인터페이스를 구성하는 사용자 정의 도구였습니다. 하지만 이 명령은 지속 구성을 구현하지 않습니다. 여러 기간에 걸쳐 네트워크 관리의 기능 추가를 위해 `ifconfig`가 향상되었습니다. 하지만 그 결과, 명령이 복잡하고 사용 시 혼동을 가져왔습니다.

인터페이스 구성 및 관리의 다른 문제는 TCP/IP 등록 정보나 조정 가능 항목을 관리하는 간단한 도구가 없다는 것입니다. `ndd` 명령은 이런 용도로 사전 설정된 사용자 정의 도구였습니다. 하지만 `ifconfig` 명령과 마찬가지로 `ndd`는 지속 구성을 구현하지 않습니다. 이전에는 부트스크립트를 편집하여 네트워크 시나리오에 대해 지속 구성을 시뮬레이션할 수 있었습니다. Oracle Solaris의 SMF(서비스 관리 기능)가 도입되면서 특히 Oracle Solaris 설치로 업그레이드할 경우 SMF 종속성 관리가 복잡하기 때문에 이러한 임시해결책의 사용이 위험할 수 있습니다.

`ipadm` 명령은 궁극적으로 인터페이스 구성에 대해 `ifconfig` 명령을 대체하도록 도입되었습니다. 또한 이 명령은 프로토콜 등록 정보를 구성하는 `ndd` 명령을 대체합니다.

인터페이스 구성 도구로서 `ipadm` 명령은 다음과 같은 이점을 제공합니다.

- 인터페이스 구성 이외의 용도로 사용되는 `ifconfig` 명령과 달리 IP 인터페이스 관리에만 사용되는 도구이므로 IP 인터페이스와 IP 주소를 보다 효율적으로 관리합니다.
- 지속 인터페이스 및 주소 구성을 구현합니다.

`ifconfig` 옵션 및 동등한 `ipadm` 하위 명령의 목록은 [부록 A, “비교 맵: `ifconfig` 및 `ipadm` 명령”](#)을 참조하십시오.

프로토콜 등록 정보를 설정하는 도구로서 `ipadm` 명령은 `ndd` 명령에 비해 다음과 같은 이점을 제공합니다.

- IP, ARP(Address Resolution Protocol), SCTP(Stream Control Transmission Protocol) 및 ICMP(Internet Control Messaging Protocol)와 TCP, UDP(User Datagram Protocol) 등의 상위 계층 프로토콜에 대한 임시 또는 지속 프로토콜 등록 정보를 설정할 수 있습니다.
- 등록 정보의 현재 및 기본값, 가능한 값 범위 등 각 TCP/IP 등록 정보에 대한 정보를 제공합니다. 따라서 디버깅 정보를 보다 쉽게 얻을 수 있습니다.
- 또한 일관된 명령 구문을 따르므로 사용하기 더 쉽습니다.

`ndd` 옵션 및 동등한 `ipadm` 하위 명령의 목록은 [부록 B, “비교 맵: `ndd` 및 `ipadm` 명령”](#)을 참조하십시오.

`ipadm` 명령에 대한 자세한 내용은 [ipadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오. `ipadm`에 사용할 수 있는 하위 명령 목록을 보려면 다음을 입력합니다.

```
# ipadm help
The following subcommands are supported:
Address          : create-addr  delete-addr  disable-addr
                  down-addr     enable-addr  refresh-addr
                  reset-addrprop set-addrprop show-addr
                  show-addrprop up-addr
Interface        : disable-if   enable-if    reset-ifprop
                  set-ifprop    show-if      show-ifprop
IP interface     : create-ip   delete-ip
IPMP interface   : add-ipmp   create-ipmp  delete-ipmp
                  remove-ipmp
Protocol property : reset-prop  set-prop     show-prop
VNI interface    : create-vni delete-vni
For more info, run: ipadm help <subcommand>.
```

## 네트워크에 대한 시스템 구성

이 장에서는 고정된 네트워크 구성을 사용하는 시스템에서 IP 인터페이스를 구성하기 위해 따르는 절차를 제공합니다. 다음 항목을 다룹니다.

- 15 페이지 “네트워크 구성(작업 맵)”
- 22 페이지 “기타 네트워크 구성 및 관리 작업”

### 네트워크 구성(작업 맵)

이 절에서는 IP 인터페이스의 기본 구성 절차에 대해 설명합니다. 다음 표에서는 구성 작업을 설명하고 이러한 작업을 해당 절차에서 매핑할 수 있습니다.

작업	설명	수행 방법
시스템에서 고유한 MAC 주소를 지원하도록 구성합니다.	SPARC 기반 시스템에서 인터페이스에 고유한 MAC 주소를 허용하도록 구성합니다.	16 페이지 “SPARC: 각 인터페이스의 MAC 주소가 고유한지 확인하는 방법”
시스템에서 활성 상태인 NCP 확인	시스템의 활성 NCP를 표시하고 DefaultFixed를 사용으로 설정합니다.	17 페이지 “시스템의 활성 NCP를 변경하는 방법”
ipadm 명령을 사용하여 기본 IP 인터페이스 구성을 수행합니다.	IP 인터페이스를 만들고 정적 또는 DHCP인 유효한 IP 주소를 인터페이스에 지정합니다.	18 페이지 “IP 인터페이스를 구성하는 방법”
데이터 링크를 사용자 정의합니다.	링크 등록 정보를 설정하여 데이터 링크를 추가로 사용자 정의합니다.	28 페이지 “데이터 링크 등록 정보 사용자 정의”
IP 인터페이스를 사용자 정의합니다.	인터페이스 등록 정보를 설정하여 IP 인터페이스를 추가로 사용자 정의합니다.	41 페이지 “IP 인터페이스 등록 정보 설정”

작업	설명	수행 방법
IP 주소를 사용자 정의합니다.	주소 등록 정보를 설정하여 IP 주소를 추가로 사용자 정의합니다.	42 페이지 “IP 주소 등록 정보 설정”
프로토콜을 사용자 정의합니다.	프로토콜 등록 정보를 설정하여 프로토콜을 추가로 사용자 정의합니다.	43 페이지 “TCP/IP 프로토콜 등록 정보 설정”
무선 네트워크를 구성합니다.	무선 네트워킹을 사용하여 네트워크에 램탐을 연결합니다.	5 장, “Oracle Solaris 실행 램탐에서 무선 네트워킹 구성”

## ▼ SPARC: 각 인터페이스의 MAC 주소가 고유한지 확인하는 방법

모든 SPARC 기반 시스템에는 시스템 차원의 MAC 주소가 있으며, 기본적으로 모든 인터페이스가 이 주소를 사용합니다. 하지만 일부 응용 프로그램에서는 호스트의 각 인터페이스가 고유한 MAC 주소를 가져야 합니다. 링크 집계 및 IPMP(IP 다중 경로 지정)와 같은 특정 유형의 인터페이스 구성에서는 인터페이스가 고유한 MAC 주소를 포함해야 합니다.

EEPROM 매개변수 `local-mac-address?`는 SPARC 기반 시스템의 모든 인터페이스가 시스템 차원의 MAC 주소를 사용하는지 또는 고유한 MAC 주소를 사용하는지를 확인합니다. 다음 절차에서는 `eeprom` 명령을 사용하여 `local-mac-address?`의 현재 값을 확인하고 필요한 경우 변경하는 방법을 설명합니다.

### 1 관리자로 전환합니다.

자세한 내용은 [Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”](#)을 참조하십시오.

### 2 시스템의 모든 인터페이스가 현재 시스템 차원의 MAC 주소를 사용하는지 확인합니다.

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

이 예에서 `eeprom` 명령에 대한 응답 `local-mac-address?=false`는 모든 인터페이스가 시스템 차원의 MAC 주소를 사용함을 나타냅니다. 인터페이스가 IPMP 그룹의 구성원이 되려면 먼저 `local-mac-address?=false` 값을 `local-mac-address?=true`로 변경해야 합니다. 또한 링크 집계에 대해서도 이 항목을 변경해야 합니다.

### 3 필요한 경우 `local-mac-address?` 값을 다음과 같이 변경합니다.

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

6단계에서 시스템을 재부트하면 출하시 설치된 MAC 주소가 있는 인터페이스가 시스템 차원의 MAC 주소 대신 이러한 출하시 설정을 사용합니다. 출하시 설치된 MAC 주소가 없는 인터페이스는 계속해서 시스템 차원의 MAC 주소를 사용합니다.

#### 4 시스템에 있는 모든 인터페이스의 MAC 주소를 확인합니다.

여러 인터페이스가 동일한 MAC 주소를 가진 경우를 찾습니다. 이 예에서는 두 인터페이스가 시스템 차원의 MAC 주소인 8:0:20:0:0:1을 사용합니다.

```
# dladm show-linkprop -p mac-address
LINK  PROPERTY  PERM VALUE                DEFAULT                POSSIBLE
net0   mac-address rw  8:0:20:0:0:1           8:0:20:0:0:1         --
net1   mac-address rw  8:0:20:0:0:1           8:0:20:0:0:1         --
net3   mac-address rw  0:14:4f:45:c:2d       0:14:4f:45:c:2d      --
```

주- 두 개 이상의 네트워크 인터페이스가 동일한 MAC 주소를 가진 경우에만 다음 단계를 계속합니다. 그렇지 않으면 마지막 단계를 진행합니다.

#### 5 필요한 경우 모든 인터페이스가 고유한 MAC 주소를 갖도록 나머지 인터페이스를 수동으로 구성합니다.

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

이전 단계의 예에서는 로컬에서 관리되는 MAC 주소를 사용하여 net0 및 net1을 구성해야 합니다. 예를 들어, 로컬에서 관리되는 MAC 주소 06:05:04:03:02를 사용하여 net0을 재구성하려면 다음 명령을 입력합니다.

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

이 명령에 대한 자세한 내용은 [dladm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

#### 6 시스템을 재부트합니다.

## ▼ 시스템의 활성화 NCP를 변경하는 방법

시스템에 사용으로 설정된 NCP 유형에 따라 시스템의 네트워크 구성이 반응적 또는 고정적인지가 결정됩니다. 반응적 구성의 시스템은 고정된 네트워크 구성과 다르게 작동합니다. 이 설명서의 모든 절차는 활성화 NCP에 적용되는 지속 구성을 만듭니다. 따라서 어떠한 절차를 수행하더라도 먼저 구성을 올바른 프로파일에 적용할 수 있도록 활성화 상태인 NCP를 확인해야 합니다. 그러면 절차를 완료한 후 시스템의 네트워크 구성이 예상한 대로 작동합니다.

#### 1 시스템의 프로파일을 나열합니다.

```
# netadm list
TYPE      PROFILE          STATE
ncp       DefaultFixed    online
ncp       Automatic        disabled
loc       Automatic        offline
loc       NoNet            offline
loc       User             offline
loc       DefaultFixed    online
```

상태가 온라인으로 나열된 프로파일이 시스템에서 활성화 NCP입니다.

시스템에서 NCP에 대한 자세한 내용을 보려면 `netadm` 명령에 `-x` 옵션을 사용합니다.

```
netadm list -x
TYPE          PROFILE          STATE          AUXILIARY STATE
ncp           DefaultFixed    online         active
ncp           Automatic       disabled      disabled by administrator
loc           Automatic       offline       conditions for activation are unmet
loc           NoNet           offline       conditions for activation are unmet
loc           User            offline       conditions for activation are unmet
loc           DefaultFixed    online         active
```

- 2 프로파일 유형 간에 전환하려면(예: 반응적 프로파일에서 고정된 프로파일로 전환) 다음 명령을 입력합니다.

```
# netadm enable -p ncp NCP-name
```

여기서 *NCP-name*은 NCP 유형의 이름입니다.

예를 들어, 시스템의 네트워크 구성이 반응적이라고 가정해보십시오. 이 설명서의 절차로 생성된 구성을 DefaultFixed NCP에 적용하려면 다음을 입력합니다.

```
# netadm enable -p ncp defaultfixed
```



주의 - 활성 프로파일을 전환하면 기존 네트워크 구성이 제거되고 새 구성이 생성됩니다. 이전 활성 NCP에서 구현된 지속 구성은 새 활성 NCP에서 제외됩니다.

## ▼ IP 인터페이스를 구성하는 방법

다음 절차에서는 시스템의 IP 인터페이스를 구성하기 위해 사용하는 기본 단계를 제공합니다.

시작하기 전에 구성을 올바른 프로파일에 적용 중인지 확인하기 위해서는 시스템에서 활성 상태인 NCP를 확인하십시오.

- 1 관리자로 전환합니다.

자세한 내용은 [Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”](#)을 참조하십시오.

- 2 인터페이스를 만듭니다.

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

*interface-class* 만들 수 있는 세 가지 인터페이스 클래스 중 하나를 나타냅니다.

- IP 인터페이스. 이 인터페이스 클래스는 네트워크 구성을 수행할 때 만드는 가장 일반적인 인터페이스입니다. 이 인터페이스 클래스를 만들려면 `create-ip` 하위 명령을 사용합니다.

- STREAMS 가상 네트워크 인터페이스 드라이버(VNI 인터페이스). 이 인터페이스 클래스를 만들려면 `create-vni` 하위 명령을 사용합니다. VNI 장치 또는 인터페이스에 대한 자세한 내용은 [vni\(7d\)](#) 매뉴얼 페이지를 참조하십시오.
- IPMP 인터페이스. 이 인터페이스는 IPMP 그룹을 구성할 때 사용됩니다. 이 인터페이스 클래스를 만들려면 `create-ipmp` 하위 명령을 사용합니다. IPMP 그룹에 대한 자세한 내용은 [Oracle Solaris 11.1 네트워크 성능 관리의 5 장, “IPMP 소개”](#)를 참조하십시오.

*interface*

인터페이스의 이름을 나타냅니다. 이 이름은 인터페이스를 만들 데이터 링크의 이름과 같습니다. 시스템의 데이터 링크를 확인하려면 `dladm show-link` 명령을 사용합니다.

### 3 다음 명령 중 하나를 선택하여 유효한 IP 주소로 IP 인터페이스를 구성합니다.

- 정적 주소를 구성하려면 다음을 입력합니다.

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

`-a address` 인터페이스에 구성할 IP 주소를 지정합니다.

---

주-터널 구성을 위해서는 일반적으로 터널 인터페이스에 대한 두 주소(로컬 주소 및 원격 주소)가 필요합니다. 터널 구성과 로컬 및 원격 주소에 대한 자세한 내용은 [Oracle Solaris 11.1 네트워크 구성 및 관리의 6 장, “IP 터널 구성”](#)을 참조하십시오.

---

숫자 IP 주소의 경우 CIDR 표기법을 사용합니다. CIDR 표기법을 사용하지 않으면 `name-service/switch` 서비스에서 `netmask`에 대해 나열된 시퀀스에 따라 또는 클래스 주소 의미 체계를 사용하여 넷마스크가 계산됩니다.

선택적으로 숫자 IP 주소 대신 호스트 이름을 지정할 수 있습니다. `/etc/hosts` 파일에서 해당 호스트 이름에 대해 숫자 IP 주소가 정의된 경우 호스트 이름을 사용해도 유효합니다. 파일에 숫자 IP 주소가 정의되지 않은 경우 `name-service/switch` 서비스에서 `host`에 대해 지정된 분석기 순서를 사용하여 숫자 값을 고유하게 가져옵니다. 지정된 호스트 이름에 대한 항목이 여러 개 있으면 오류가 생성됩니다.

주 - 부트 프로세스 도중 IP 주소가 먼저 생성된 후 이름 지정 서비스가 온라인 상태로 전환됩니다. 따라서 네트워크 구성에서 사용된 호스트 이름이 `/etc/hosts` 파일에 정의되어 있는지 확인해야 합니다.

[*interface* | *addrobj*]

Oracle Solaris에서 각 주소는 해당 주소 객체로 식별되며 명령에 *addrobj*로 표시됩니다. 주소에 대한 이후 구성을 위해서는 실제 IP 주소 대신 주소 객체를 참조합니다. 예를 들어, `ipadm show-addr addrobj` 또는 `ipadm delete-addr addrobj`를 입력합니다. 주소 객체 이름을 자동으로 만들려면 *interface*에 대한 인터페이스 이름을 지정합니다. 그렇지 않으면 주소 객체 이름을 직접 제공합니다.

- 인터페이스 이름을 지정한 경우 주소 객체 이름이 자동으로 *interface/address-family* 형식으로 바뀝니다. *Address family*는 IPv4 주소의 경우 `v4`이고 IPv6 주소의 경우 `v6`입니다. 동일 인터페이스에서 다중 주소는 주소 객체 이름에 연결된 영문자가 포함됩니다(예: `net0/v4`, `net0/v4a`, `net0/v4b`, `net0/v6`, `net0/v6a` 등).
- *addrobj*에 대한 주소 객체 이름을 수동으로 바꿀 경우에는 *interface/user-specified-string* 형식을 사용해야 합니다. *User-specified-string*은 영문자로 시작하고 최대 길이가 32자인 영숫자의 문자열을 나타냅니다. 예를 들어, `net0/static`, `net0/static1`, `net1/private` 등으로 주소 객체 이름을 지정할 수 있습니다.
- 비정적 주소를 구성하려면 다음을 입력합니다.

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

여기서 *address-type*은 `dhcp` 또는 `addrconf`입니다. `Addrconf`는 자동으로 생성된 IPv6 주소를 나타냅니다.

[*interface* | *addrobj*]에 대한 자세한 내용은 정적 주소 만들기에 대한 이전 설명을 참조하십시오.

#### 4 (선택 사항) 새로 구성된 IP 인터페이스에 대한 정보를 표시합니다.

확인하려는 정보에 따라 다음 명령을 사용할 수 있습니다.

```
# ipadm [interface]
```

*interface*를 지정하지 않으면 시스템의 모든 인터페이스에 대한 정보가 표시됩니다.

`ipadm show-*` 하위 명령의 출력 결과에 대한 자세한 내용은 50 페이지 “IP 인터페이스 및 주소 모니터링”을 참조하십시오.

- 5 호스트 이름을 사용하는 정적 IP 주소를 구성할 경우 `/etc/hosts` 파일에서 IP 주소에 대한 항목을 추가합니다.

이 파일의 항목은 IP 주소와 해당 호스트 이름으로 구성됩니다.

---

주-DHCP 주소를 구성하는 경우 `/etc/hosts` 파일을 업데이트할 필요가 없습니다.

---

- 6 기본 경로를 정의합니다.

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

`netstat -r` 명령으로 경로 지정 테이블의 내용을 확인할 수 있습니다.

메시징 경로에 대한 자세한 내용은 `route(1M)` 및 `routed(1M)` 매뉴얼 페이지를 참조하십시오. 또한 **Oracle Solaris 11.1 네트워크 구성 및 관리**의 “경로 지정 테이블 및 경로 지정 유형”도 참조하십시오.

## 예 2-1 정적 IP 주소를 사용하여 네트워크 인터페이스 구성

이 예제에서는 정적 IP 주소를 사용하여 인터페이스를 구성하는 방법에 대해 설명합니다. 이 예제에서는 먼저 사용자가 고정된 네트워크 구성에 대해 `dladm` 및 `ipadm` 명령을 사용할 수 있도록 허용하기 위해 시스템에서 `DefaultFixed NCP`를 사용으로 설정합니다.

```
# 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	BRIDGE	OVER
net3	phys	1500	up	--	--

```
# ipadm create-ip net3
```

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

```
ipadm: net3/v4
```

```
# ipadm
```

NAME	CLASS/TYPE	STATE	UNDER	ADDR
lo0	loopback	ok	--	--
lo0/v4	static	ok	--	127.0.0.1/8
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 campus01
```

```
# route -p add default 192.168.84.1
```

```
# netstat -r
```

```
Routing Table: IPv4
Destination      Gateway          Flags Ref    Use      Interface
-----
default          some.machine.com UG      2      10466
192.168.84.0    192.168.84.3   U       3      1810 net3
localhost       localhost       UH      2       12 lo0
```

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

campus01이 /etc/hosts 파일에서 이미 정의된 경우 다음 주소를 지정할 때 해당 호스트 이름을 사용할 수 있습니다.

```
# ipadm create-addr -a campus01 net3
ipadm: net3/v4
```

## 예 2-2 IP 주소를 사용하여 자동으로 네트워크 인터페이스 구성

이 예제에서 IP 인터페이스는 DHCP 서버에서 해당 주소를 수신하도록 구성됩니다.

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

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

```
# ipadm create-ip net3
# ipadm create-addr -T dhcp net3
ipadm: net3v4
```

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

## 기타 네트워크 구성 및 관리 작업

이 설명서에서는 시스템을 네트워크에 연결하는 기본 네트워크 구성에 대해 설명합니다. 특히 시스템의 데이터 링크 및 인터페이스 구성에 대한 정보를 집중적으로 다룹니다. 기타 네트워크 구성 및 관리 작업은 다른 네트워킹 설명서에 설명된 대로 수행할 수 있습니다. 시스템이 고정된 네트워크로 구성되었다고 가정하면 다음과 같은 기타 작업에 대해 다음 설명서를 참조할 수 있습니다.

- 시스템을 라우터, 네트워크 구성 서버 등으로 구성하려면 **Oracle Solaris 11.1 네트워크 구성 및 관리**를 참조하십시오.
- 고급 데이터 링크 및 IP 인터페이스 구성을 수행하려면 **Oracle Solaris 11.1 네트워크 성능 관리**를 참조하십시오.
- 기본 구성을 작성하고 링크 집계, IPMP 그룹 등의 구성과 같은 네트워크 성능을 향상시키려면 **Oracle Solaris 11.1 네트워크 성능 관리**를 참조하십시오.
- 네트워크에 대한 보안을 설정하려면 **Oracle Solaris 11.1의 네트워크 보안**을 참조하십시오.
- 네트워크 가상화를 구현하기 위해서는 **Oracle Solaris 11.1에서 가상 네트워크 사용**을 참조하십시오.

DHCP, 이름 서비스 등과 같이 특정 네트워킹 영역을 전문으로 다루는 다른 설명서도 라이브러리에 제공됩니다.



## 데이터 링크 작업

---

이 장에서는 dladm 명령에 대해 설명하고 데이터 링크에서 이 명령을 사용하여 현재 구성을 표시하거나, 해당 속성의 기본값을 변경하거나 시스템에서 데이터 링크를 삭제하는 방법에 대해 설명합니다. 다음 항목을 다룹니다.

- 25 페이지 “기본 dladm 명령”
- 28 페이지 “데이터 링크 등록 정보 사용자 정의”
- 33 페이지 “dladm 명령을 사용한 기타 구성 작업”

### 기본 dladm 명령

이 절에서는 시스템의 데이터 링크에서 일반적으로 사용하는 기본 dladm 명령에 대해 설명합니다. 이 절에 나열된 것 이외의 추가 dladm 하위 명령도 지원됩니다. 다른 하위 명령을 보려면 [dladm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

---

주 - 데이터 링크 정보를 표시하는 dladm 하위 명령을 제외하고 다른 모든 하위 명령을 사용하려면 먼저 데이터 링크에 대한 기존 인터페이스 구성을 제거해야 합니다. IP 인터페이스 구성을 제거하려면 [39 페이지](#) “IP 인터페이스 구성 제거(ipadm delete-ip)”를 참조하십시오.

---

### 데이터 링크에 대한 일반 정보 표시(dladm)

dladm 명령을 단독으로 사용하면 시스템 클래스, 상태, 기본 물리적 링크를 포함하여 시스템의 데이터 링크에 대한 일반 정보가 표시됩니다.

```
# 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
```

데이터 링크는 링크 집계, VLAN(가상 LAN) 또는 VNIC(가상 NIC)와 같은 물리적 링크가 아닌 다른 클래스일 수 있습니다. 이러한 다른 데이터 링크도 dladm 명령으로 표시되는 기본 정보에 포함됩니다. 예를 들어, 출력에 net1 및 net4 물리적 데이터 링크를 통해 구성된 링크 집계 aggr0이 표시됩니다.

링크 집계 및 VLAN에 대한 자세한 내용은 [Oracle Solaris 11.1 네트워크 성능 관리](#)를 참조하십시오. VNIC에 대한 자세한 내용은 [Oracle Solaris 11.1에서 가상 네트워크 사용](#)을 참조하십시오.

## 시스템 데이터 링크 표시(dladm show-link)

시스템의 데이터 링크를 표시하려면 dladm show-link를 사용합니다. 시스템에는 설치된 NIC만큼 많은 데이터 링크가 포함됩니다. 이 명령의 옵션을 사용하여 표시되는 정보를 사용자 정의할 수 있습니다. 예를 들어, -p 옵션을 사용하면 데이터 링크에 대한 지속 구성 정보가 포함됩니다. 이 명령으로 제공되는 정보에 따라 추가 네트워크 구성을 계속할 수 있습니다. 예를 들어, 시스템의 NIC 개수를 확인하고 IP 인터페이스를 구성하는 데 사용할 데이터 링크를 선택할 수 있습니다.

명령을 실행하면 다음과 유사한 정보가 표시됩니다.

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

이 예제에서는 해당 물리적 NIC와 직접적으로 연관된 데이터 링크가 시스템에 세 개 있음을 보여줍니다. phys 클래스 아래에 데이터 링크로 구성된 집계 또는 가상 NIC와 같은 특수한 데이터 링크는 존재하지 않습니다.

## 데이터 링크의 물리적 속성 표시(dladm show-phys)

데이터 링크와 연결된 물리적 NIC와 관련하여 시스템 데이터 링크에 대한 정보를 가져오려면 dladm show-phys를 사용합니다. 이 명령을 옵션 없이 사용하면 다음과 유사한 정보가 표시됩니다.

```
# 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
```

출력에는 다른 세부 정보 외에도 일반 링크 이름의 데이터 링크가 연결된 물리적 NIC가 표시됩니다. 예를 들어, `net0`은 NIC `e1000g0`의 데이터 링크 이름입니다. 데이터 링크에 대해 설정된 플래그에 대한 자세한 내용을 보려면 `-p` 옵션을 사용합니다. 예를 들어, `r` 플래그 지정된 데이터 링크는 기본 NIC가 제거되었음을 의미합니다.

이 명령에서 유용한 또 다른 옵션은 각 데이터 링크의 물리적 위치를 보여주는 `-L`입니다. 이 위치에 따라 `net0`, `net1` 등과 같은 데이터 링크의 인스턴스 번호가 결정됩니다.

```
# 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
```

## 데이터 링크 삭제(dladm delete-phys)

시스템에서 데이터 링크를 제거하려면 `dladm delete-phys`를 사용합니다.

데이터 링크 제거는 물리적 NIC 제거와 느슨한 형태로만 연결되어 있습니다. 예를 들어, 물리적 NIC가 시스템에서 제거되었다고 가정해보십시오. **Oracle Solaris 11 네트워킹 소개의 “Oracle Solaris의 네트워킹 스택”**에 설명된 대로 소프트웨어 층이 하드웨어 층에 더 이상 바인딩되지 않기 때문에 이 NIC와 연관된 데이터 링크 구성은 남은 상태로 있습니다. 따라서 이 데이터 링크의 이름을 다른 NIC에 연결된 링크에 지정하여 다른 기본 물리적 NIC에서 이 데이터 링크 구성을 사용할 수 있습니다.

NIC를 교체하지 않고 분리했으며, 해당 데이터 링크 구성이 더 이상 필요하지 않으면 다음과 같이 데이터 링크를 삭제할 수 있습니다.

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

---

참고 - 데이터 링크의 NIC가 제거되었는지 여부를 확인하려면 `dladm show-phys -P` 명령을 사용합니다.

---

## 데이터 링크 이름 바꾸기(dladm rename-link)

데이터 링크 이름을 바꾸려면 `dladm rename-link`를 사용합니다. Oracle Solaris 11 시스템에서 OS는 모든 데이터 링크에 일반 이름을 자동으로 제공합니다. 일반 데이터 링크 이름은 **Oracle Solaris 11 네트워킹 소개의 “기본 일반 링크 이름”**에 설명되어 있습니다.

기본적으로 이러한 일반 이름은 `net0`, `net1`, `net2` 등과 같이 `net n`의 이름 지정 형식을 사용합니다. 이름은 OS에서 관리되므로 사용자가 일반적인 관리 작업의 일부로 데이터 링크 이름을 바꿀 필요가 없습니다. 링크 이름을 바꿔야 하는 절차에 대해서는 [34 페이지 “시스템에서 주 인터페이스를 전환하는 방법”](#)을 참조하십시오.

## 데이터 링크 등록 정보 사용자 정의

기본 데이터 링크 구성을 수행하는 것은 물론 `dladm` 명령을 사용하여 데이터 링크 등록 정보를 설정하고 네트워크 요구 사항에 따라 사용자 정의를 할 수도 있습니다.

데이터 링크 등록 정보에는 세 가지 `dladm` 하위 명령이 사용됩니다.

- `dladm show-linkprop [-p property] [datalink]`는 데이터 링크의 등록 정보와 현재 값을 표시합니다. `-p property` 옵션을 사용하지 않으면 데이터 링크의 모든 등록 정보가 나열됩니다. 데이터 링크를 지정하지 않으면 모든 데이터 링크의 모든 등록 정보가 나열됩니다.
- `dladm set-linkprop -p property =value datalink`는 데이터 링크의 등록 정보에 값을 지정합니다.
- `dladm reset-linkprop -p property datalink`는 특정 등록 정보를 해당 기본값으로 재설정합니다.

## 데이터 링크 등록 정보 개요

사용자 정의를 할 수 있는 데이터 링크 등록 정보는 특정 NIC 드라이버가 지원하는 등록 정보에 따라 달라집니다. `dladm` 명령을 사용하여 구성할 수 있는 데이터 링크 등록 정보는 다음 두 가지 범주 중 하나에 속합니다.

- **공용 등록 정보:** 링크 속도, 이더넷의 자동 협상 또는 모든 데이터 링크 드라이버에 적용할 수 있는 MTU(최대 전송 단위) 크기 등 지정된 매체 유형의 드라이버에 적용할 수 있습니다.
- **개인 등록 정보:** 지정된 매체 유형에 대한 NIC 드라이버의 특정 부분에 따라 고유합니다. 이러한 등록 정보는 드라이버와 연결된 하드웨어 또는 드라이버 구현 자체의 세부 정보(예: 디버깅 관련 조정 가능 항목)와 긴밀한 관계가 있기 때문에 이 부분에만 관련된 것일 수 있습니다.

링크 등록 정보에는 일반적으로 세 가지 기본값이 포함됩니다. 하지만 특정 네트워킹 시나리오에서는 특정 등록 정보 값을 변경해야 할 수도 있습니다. 예를 들어, NIC가 자동 협상을 제대로 수행하지 않는 이전 스위치와 통신 중일 수 있습니다. 또는 스위치가 점포 프레임의 지원하도록 구성되었을 수도 있습니다. 또는 패킷 전송 또는 패킷 수신을 규제하는 드라이버별 등록 정보를 특정 드라이버에 맞게 수정해야 할 수도 있습니다. 다음 절에서는 선택한 등록 정보에 대해 설명하고 네트워크 환경에서 작동하도록 해당 값을 변경하는 방법에 대해 설명합니다.

## 점포 프레임 지원 사용

MTU는 프로토콜이 시스템에서 전송할 수 있는 최대 패킷 크기를 정의합니다. 기본적으로 대부분의 NIC 드라이버는 MTU 크기를 1500으로 정의합니다. 하지만 점포 프레임이 네트워크를 통과할 경우 기본값이 충분하지 않습니다. 점포 프레임은 지원하려면 MTU 크기가 적어도 9000 이상이어야 합니다.

MTU 크기를 기본값에서 변경하려면 다음 명령을 입력합니다.

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

MTU 크기를 변경한 후에는 데이터 링크에 대한 IP 인터페이스를 재구성할 수 있습니다.

다음 예제에서는 점보 프레임에 대한 지원을 사용으로 설정하기 위한 단계를 보여줍니다. 이 예제에서는 데이터 링크에 대한 기존 IP 인터페이스 구성을 이미 제거했다고 가정합니다.

```
# dladm show-linkprop -p mtu net1
LINK      PROPERTY  VALUE    DEFAULT  POSSIBLE
net1      mtu       1500    1500    --
# dladm set-linkprop -p mtu=9000 net1
# dladm show-link web1
LINK      CLASS     MTU      STATE    BRIDGE   OVER
web1      phys     9000    up       --       --
```

## 링크 속도 매개변수 수정

대부분의 네트워크 설정은 다양한 속도 기능의 시스템 조합으로 구성됩니다. 각 시스템은 네트워크의 다른 시스템에 각 시스템이 네트워크 트래픽을 전송 및 수신할 수 있는 속도 기능을 알립니다. 다음 두 가지 데이터 링크 등록 정보는 시스템이 알리는 속도 기능을 규제합니다.

- 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

각 링크 속도 기능은 알려진 속도(adv\*\_cap)와 사용으로 설정된 알려진 속도(en\*\_cap)의 등록 정보 쌍으로 참조됩니다. 또한 데이터 링크 속도 정보는 등록 정보 이름에서 \*fdx\* 및 \*hdx\*로 지정된 대로 전이중 기능과 반이중 기능 모두에 제공됩니다. 알려진 속도 등록 정보는 특정 데이터 링크 속도가 알려졌는지 여부를 나타내는 읽기 전용 등록 정보입니다. 특정 데이터 링크 속도를 알릴지 여부는 해당 en\*\_cap 등록 정보를 설정하여 결정합니다.

기본적으로 데이터 링크의 모든 속도와 이중 기능이 알려집니다. 하지만 새로운 시스템이 이전 시스템과 통신하고 자동 협상이 사용 안함으로 설정되었거나 지원되지 않는 경우도 존재할 수 있습니다. 이러한 두 시스템 간의 통신을 사용으로 설정하기 위해서는 이전 시스템과 최신 시스템 간에 알려진 속도를 하위 값으로 변경해야 할 수 있습니다. 시스템의 기가비트 기능을 해제하고 느린 속도 기능만 알려야 할 수 있습니다. 이 경우 전이중 기능 및 반이중 기능 모두에 대해 다음을 입력해야 합니다.

```
# dladm set-linkprop -p en_100fdx_cap=0 datalink
# dladm set-linkprop -p en_100hdx_cap=0 datalink
```

이 명령은 전이중 기능 및 반이중 기능 모두에 대해 시스템의 기가비트 기능 알림을 해제합니다.

이러한 등록 정보의 새 값을 표시하려면 `dladm show-linkprop` 명령을 사용합니다.

```
# dladm show-linkprop -p adv_10gfdx_cap datalink
# dladm show-linkprop -p adv_100hdx_cap datalink
```

일반적으로 사용으로 설정된 지정된 속도 등록 정보와 알려진 해당 등록 정보의 값은 동일합니다. 하지만 NIC가 전원 관리 등의 고급 기능을 지원하는 경우 해당 기능이 호스트와 링크 파트너 간에 실제로 알려지는 비트에 제한을 설정할 수도 있습니다. 예를 들어, 전원 관리를 사용할 경우 `adv_*_cap` 등록 정보의 설정이 `en_*_cap` 등록 정보 설정의 일부일 뿐입니다.

## 데이터 링크에 STREAMS 모듈 설정

데이터 링크를 열 때 스트림에 최대 8개의 STREAMS 모듈이 푸시되도록 설정할 수 있습니다. 일반적으로 이러한 모듈은 VPN(가상 사설망) 및 방화벽과 같은 타사 네트워크 소프트웨어에 사용됩니다. 네트워킹 소프트웨어에 대한 설명서는 소프트웨어 공급업체가 제공합니다.

특정 데이터 링크에 푸시할 모듈 목록은 `autopush` 등록 정보로 제어됩니다. `autopush` 등록 정보의 값은 `dladm set-linkprop` 하위 명령을 사용하여 설정됩니다.

또한 별도의 `autopush` 명령을 사용하여 드라이버별로 데이터 링크의 스트림에 모듈을 푸시할 수도 있습니다. 이 명령은 각 드라이버에 대해 설정되었고 푸시할 모듈을 명령에 알려 주는 구성 파일을 사용합니다. 하지만 드라이버는 항상 NIC에 바인딩됩니다. 데이터 링크의 기본 NIC를 제거하면 링크의 `autopush` 등록 정보에 대한 정보도 손실됩니다.

따라서 `dladm` 명령은 `autopush` 명령보다 이 목적에 적합한 도구입니다. 특정 데이터 링크에 대해 드라이버별 및 링크별 `autopush` 구성 유형이 있는 경우 `dladm set-linkprop`으로 설정된 링크별 정보가 사용되며 드라이버별 정보는 무시됩니다.

데이터 링크가 열릴 때 STREAMS에 모듈을 푸시하려면 동일한 `dladm set-linkprop` 명령을 사용하여 `autopush` 등록 정보에 대한 모듈을 지정합니다. 예를 들어, `vpnmod` 및 `bufmod` 모듈을 `net0` 링크 위에 푸시하려면 다음을 입력합니다.

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

## 직접 메모리 액세스 바인딩을 사용하도록 e1000g 드라이버 설정

이 절 및 다음 절에서는 개인 등록 정보를 구성하는 방법을 보여줍니다. 두 절 모두 e1000g 드라이버와 관련된 등록 정보에 적용됩니다. 하지만 이러한 절의 일반 정보는 다른 NIC 드라이버의 개인 등록 정보를 구성할 때 적용됩니다.

파일 전송 등의 대량 트래픽이 있을 경우 대체로 네트워크에서 큰 패킷을 협상해야 합니다. 이 경우 패킷 조각 크기에 대해 임계값이 정의되는 DMA(직접 메모리 액세스) 바인딩을 자동으로 사용하도록 구성하면 e1000g 드라이버의 성능을 향상시킬 수 있습니다. 조각 크기가 임계값을 초과할 경우 DMA 바인딩이 패킷 전송에 사용됩니다. 조각 크기가 임계값 이내인 경우 조각 데이터가 미리 할당된 전송 버퍼에 복사되는 bcopy 모드가 사용됩니다.

```
# dladm set-linkprop -p _tx_bcopy_threshold=value datalink
```

이 등록 정보에 유효한 임계값 범위는 60에서 2048 사이입니다.

---

주 - 모든 데이터 링크 이름은 자동으로 일반 이름으로 지정됩니다. 기본 NIC가 e1000g인 데이터 링크에 이 개인 등록 정보가 구성되어 있는지 확인해야 합니다. 등록 정보를 설정하기 전 `dladm show-phys`를 사용하여 확인하십시오.

공용 등록 정보를 구성할 때와 마찬가지로 개인 등록 정보 값을 수정하려면 먼저 IP 인터페이스를 삭제해야 합니다.

---

다음과 유사한 단계를 수행해야 할 수 있습니다.

```
# dladm show-phys
LINK  MEDIA  STATE  SPEED  DUPLEX  DEVICE
net0  Ethernet  up     100Mb  full    nge0
net1  Ethernet  up     100Mb  full    e1000g0

# dladm set-linkprop -p _tx_bcopy_threshold=1024 net1
```

## 인터럽트 속도를 수동으로 설정

인터럽트가 e1000g 드라이버에 의해 전달되는 속도를 규제하는 등록 정보는 네트워크 및 시스템 성능에도 영향을 줍니다. 일반적으로 네트워크 패킷은 각 패킷에 대해 인터럽트를 생성하여 스택의 상위 계층으로 전달됩니다. 기본적으로 인터럽트 속도는 커널의 GLD 계층에서 자동으로 조정됩니다. 하지만 이 모드가 모든 네트워크 트래픽 상태에서 바람직한 것은 아닙니다. 이 문제에 대한 자세한 내용은 1996년 USENIX 기술 회의에서 발표된 문서(<http://www.stanford.edu/class/cs240/readings/mogul.pdf>)를 참조하십시오. 따라서 성능 향상을 위해 수동으로 인터럽트 속도를 설정해야 하는 경우도 있습니다.

인터럽트 속도를 정의하려면 다음 등록 정보를 설정합니다.

- `_intr_throttling_rate`는 네트워크 트래픽 상태에 관계없이 인터럽트 검증 간의 지연을 결정합니다.
- `_intr_adaptive`는 인터럽트 제한 속도의 자동 조정을 사용으로 설정할지 여부를 결정합니다. 기본적으로 이 등록 정보는 사용으로 설정됩니다.

먼저 인터럽트 제한 속도의 자동 조정을 해제합니다. 그런 후 인터럽트 제한 속도 등록 정보를 수동으로 설정합니다.

e1000g NIC의 인터럽트 제한 속도를 수정해야 하는 x86 기반 시스템이 있다고 가정해보십시오. 또한 e1000g0의 데이터 링크 이름은 net1입니다. 이 경우 다음과 같은 명령을 입력해야 합니다.

```
# dladm set-linkprop -p _intr_adaptive=0 net1
# dladm set-linkprop -p _intr-throttling_rate=1024 net1
```

## 데이터 링크 등록 정보에 대한 상태 정보 가져오기

데이터 링크 등록 정보에 대한 정보를 가져오기 위해서는 다음 명령 중 하나를 사용할 수 있습니다.

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

### 데이터 링크 등록 정보 표시(dladm show-linkprop)

이 방식은 28 페이지 “데이터 링크 등록 정보 사용자 정의”에 설명되어 있습니다. 데이터 링크 등록 정보의 전체 목록을 표시하려면 등록 정보를 지정하지 않고 명령을 입력합니다. 예를 들면 다음과 같습니다.

```
# dladm show-linkprop net1
LINK      PROPERTY      VALUE      DEFAULT    POSSIBLE
net1     speed         1000      --         --
net1     autopush     --        --         --
net1     zone         --        --         --
net1     duplex       half      --         half,full
net1     state        unknown   up         up,down
net1     adv_autoneg_cap 1         1         1,0
net1     mtu          1500     1500     --
net1     flowctrl     no        bi        no,tx,rx,bi
net1     adv_1000fdx_cap 1         1         1,0
net1     en_1000fdx_cap 1         1         1,0
net1     adv_1000hdx_cap 1         1         1,0
net1     en_1000hdx_cap 1         1         1,0
net1     adv_100fdx_cap 0         0         1,0
net1     en_100fdx_cap 0         0         1,0
net1     adv_100hdx_cap 0         0         1,0
net1     en_100hdx_cap 0         0         1,0
```

```
net1    adv_10fdx_cap    0      0      1,0
net1    en_10fdx_cap     0      0      1,0
net1    adv_10hdx_cap    0      0      1,0
net1    en_10hdx_cap     0      0      1,0
```

## 이더넷 등록 정보 값 표시(dladm show-ether)

dladm show-ether 명령에 옵션을 사용하지 않으면 데이터 링크의 현재 이더넷 등록 정보 값만 표시됩니다. 기본적으로 제공된 것 이상의 추가 정보를 가져오려면 -x 옵션을 사용합니다. 다음은 이 명령의 사용 방법에 대한 예제입니다.

```
# 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 옵션을 사용하면 이 명령은 지정한 링크의 내장 기능뿐 아니라 호스트와 링크 파트너 간에 현재 알려진 기능도 표시합니다. 다음은 이전 예제에 표시된 정보에 대한 설명입니다.

- 현재 이더넷 장치 상태의 경우 링크가 전이중에서 초당 1기가비트 속도로 작동하고 있습니다. 자동 협상 기능이 사용으로 설정되었으며 호스트와 링크 파트너가 일시 중지 프레임을 보내고 받을 수 있는 양방향 플로우 제어가 있습니다. 이 정보는 출력의 첫번째 행에 표시됩니다.
- 이후 행에는 데이터 링크 속도 기능 및 알려진 실제 데이터 링크 속도에 대한 정보뿐만 아니라 다음과 같은 피어 시스템의 정보가 표시됩니다.
  - 이더넷 장치의 기능이 나열됩니다. 협상 유형은 자동으로 설정할 수 있습니다. 또한 장치가 전이중 및 반이중에서 초당 1기가비트, 초당 100메가비트 및 초당 10메가비트 속도를 지원할 수 있습니다. 마찬가지로, 호스트와 링크 파트너 간에 일시 중지 프레임을 양방향으로 보내거나 받을 수 있습니다.
  - net1의 기능은 자동 협상, 속도-이중 및 일시 중지 프레임의 플로우 제어로 알려집니다.
  - 이와 유사하게, net1의 링크 또는 피어 파트너도 자동 협상, 속도-이중 및 일시 중지 프레임의 플로우 제어를 알립니다.

## dladm 명령을 사용한 기타 구성 작업

이 절에서는 주 인터페이스 전환 또는 DR(동적 재구성)과 같은 dladm 명령을 사용하여 간소화된 추가 구성 절차에 대해 설명합니다.

## ▼ 시스템에서 주 인터페이스를 전환하는 방법

시스템의 주 인터페이스를 변경하는 경우는 데이터 링크 이름을 바꾸는 경우입니다. 이후 절차는 다음과 같은 시스템 구성을 기반으로 합니다.

- 시스템에 net0 및 net1의 두 데이터 링크가 포함됩니다.
- 기본 NIC는 각각 e1000g0 및 nge0입니다.
- IP 인터페이스는 net0에 구성되어 있습니다. IP 인터페이스는 항상 기본 데이터 링크의 이름을 가져옵니다.

시스템의 주 인터페이스는 0번 인스턴스 번호(0)를 기반으로 하는 net0입니다. 주 인터페이스는 e1000g0으로 구성되어 있습니다. 다음 단계에서는 nge0을 통한 데이터 링크 구성을 주 인터페이스 구성으로 만드는 방법에 대해 설명합니다.

### 1 관리자로 전환합니다.

자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”**을 참조하십시오.

### 2 시스템의 데이터 링크에 대한 물리적 속성을 표시합니다.

```
# dladm show-phys
```

### 3 주 IP 인터페이스를 삭제합니다.

```
# ipadm delete-ip interface
```

---

주-ipadm 명령에 대한 자세한 내용은 4장, “IP 인터페이스 작업” 및 ipadm(1M) 매뉴얼 페이지를 참조하십시오.

---

### 4 주 링크 이름을 시스템의 다른 데이터 링크에서 사용되지 않는 이름으로 바꿉니다.

```
# dladm rename-link primary-link unused-name
```

### 5 주장치가 되도록 지정된 데이터 링크에 주 링크 이름을 지정합니다.

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

## 예 3-1 주 인터페이스 전환

다음 예제에서는 절차의 모든 단계를 조합하여 시스템의 주 인터페이스를 변경합니다. 예제 끝에서 e1000g0으로 구성된 주 인터페이스는 nge0으로 구성된 인터페이스로 바뀝니다. 주 링크를 다른 NIC로 전환한 후에는 새 NIC의 데이터 링크를 통해 인터페이스를 구성할 수 있습니다.

```
# dladm show-phys
LINK MEDIA STATE SPEED DUPLEX DEVICE
net0 Ethernet up 100Mb full e1000g0
net1 Ethernet up 100Mb full nge0
```

```
# ipadm delete-ip net0
# dladm rename-link net0 oldnet0
# dladm rename-link net1 net0

# ipadm create-ip net0
# ipadm create-addr -a 192.168.10.10/24 net0
ipadm: net0/v4

# dladm show-phys
LINK      MEDIA      STATE  SPEED  DUPLEX  DEVICE
oldnet0   Ethernet  up     1000   full    e1000g0
net0      Ethernet  up     1000   full    nge0
```

## ▼ 동적 재구성을 사용하여 네트워크 인터페이스 카드를 교체하는 방법

이 절차는 DR(동적 재구성)을 지원하는 시스템에만 적용됩니다. 특히 DR이 완료된 후 구성 단계를 나타냅니다. Oracle Solaris 11에서는 이제 DR을 완료한 후 더 이상 네트워크 링크를 재구성할 필요가 없습니다. 대신 제거된 NIC의 링크 구성만 교체 NIC에 전송합니다.

이 절차에서는 DR 자체를 수행하는 단계에 대해 자세히 다루지 않습니다. 이러한 단계는 시스템 설명서를 참조하십시오.

DR 소개를 보려면 **Oracle Solaris 11.1 관리: 장치 및 파일 시스템의 4 장**, “동적으로 장치 구성(작업)”을 참조하십시오.

시작하기 전에 DR 수행 절차는 시스템 유형에 따라 달라집니다. 먼저 다음을 완료해야 합니다.

- 시스템이 DR을 지원하는지 확인합니다.
- 시스템의 DR에 대해 설명하는 해당 매뉴얼을 참조하십시오.

Oracle Sun 서버의 DR에 대한 현재 설명서를 찾으려면 <http://www.oracle.com/technetwork/indexes/documentation/index.html>에서 동적 재구성을 검색합니다.

Oracle Solaris Cluster 환경에서 DR 수행에 대한 자세한 내용은 **Oracle Solaris Cluster 시스템 관리 설명서**를 참조하십시오.

### 1 관리자로 전환합니다.

자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”**을 참조하십시오.

### 2 (옵션) 시스템에서 데이터 링크의 물리적 속성 및 해당 위치에 대한 정보를 표시합니다.

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

dladm show-phys -L로 표시되는 정보 유형에 대한 자세한 내용은 **dladm(1M)** 매뉴얼 페이지를 참조하십시오.

**3 시스템 설명서에 따라 DR을 수행합니다.**

이 단계를 수행하려면 시스템의 DR 설명서를 참조하십시오.  
교체 NIC를 설치한 후 다음 단계에서 계속합니다.

**4 적용되는 상황에 따라 다음 단계 중 하나를 수행합니다.**

- 이전 NIC와 동일한 슬롯에 교체 NIC를 삽입한 경우 5단계를 진행합니다.  
새 NIC가 이전 NIC에서 사용한 것과 동일한 위치를 사용하는 경우 새 NIC가 이전 NIC의 링크 이름과 구성을 상속합니다.
- 교체 NIC를 다른 슬롯에 삽입했고 새 NIC가 제거되는 NIC의 데이터 링크 구성을 상속해야 하는 경우 다음을 입력합니다.

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

*new-datalink*   이전 NIC가 제거된 위치와 다른 슬롯에 있는 교체 NIC의 데이터 링크를 참조합니다.

*old-datalink*   제거된 이전 NIC와 연결된 데이터 링크 이름을 참조합니다.

---

주 - 이 시나리오에서는 이전 NIC를 제거한 슬롯을 비워 두어야 합니다.

---

예를 들어, 슬롯 1의 NIC가 제거되었고 새 NIC는 슬롯 2에 삽입되어 있습니다. 이때, 슬롯 1에는 삽입된 NIC가 없습니다. 슬롯 1의 데이터 링크가 *net0*이라고 가정하면 슬롯 2의 데이터 링크는 *net1*입니다. 새 NIC의 데이터 링크가 이전 NIC의 데이터 링크 구성을 상속하도록 하려면 다음을 입력합니다.

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

**5 새 NIC의 리소스를 Oracle Solaris에서 사용할 수 있게 하여 DR 프로세스를 완료합니다.**

예를 들어, *cfgadm* 명령을 사용하여 NIC를 구성할 수 있습니다. 자세한 내용은 *cfgadm(1M)* 매뉴얼 페이지를 참조하십시오.

**6 (옵션) 링크 정보를 표시합니다.**

*dladm show-phys* 또는 *dladm show-link*를 사용하여 데이터 링크 정보를 표시할 수 있습니다.

**예 3-2 새 네트워크 카드를 설치하여 동적 재구성 수행**

이 예에서는 링크 이름이 *net0*인 *bge* 카드가 *e1000g* 카드로 교체되는 방식을 보여줍니다. *e1000g*가 시스템에 연결된 후 *net0*의 링크 구성이 *bge*에서 *e1000g*로 전송됩니다.

```
# 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
```

관리자는 `cfgadm`을 사용하여 `bge`를 제거하고 해당 위치에 `e1000g`를 설치하는 등의 DR 관련 단계를 수행합니다. 카드가 설치된 후 `e1000g0`의 데이터 링크는 자동으로 이름 `net0`을 갖게 되고 링크 구성을 상속합니다.

```
# 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    ---
```



## IP 인터페이스 작업

---

이 장에서는 ipadm 명령에 대해 설명하고 IP 인터페이스에서 이 명령을 사용하는 방법에 대해 설명합니다. ipadm 명령 및 해당 이점의 개요는 13 페이지 “ipadm 명령”을 참조하십시오.

다음 항목을 다룹니다.

- 39 페이지 “기본 ipadm 명령”
- 41 페이지 “IP 인터페이스 등록 정보 설정”
- 42 페이지 “IP 주소 등록 정보 설정”
- 43 페이지 “TCP/IP 프로토콜 등록 정보 설정”
- 50 페이지 “IP 인터페이스 및 주소 모니터링”

### 기본 ipadm 명령

18 페이지 “IP 인터페이스를 구성하는 방법”에서는 세 가지 주요 ipadm 하위 명령이 소개되었습니다.

- ipadm
- ipadm create-ip
- ipadm create-addr

이 절에서는 IP 인터페이스에서 ipadm 명령의 다른 선택적인 사용에 대해 설명합니다. 이 목록은 전체 목록이 아닙니다. ipadm 명령과 사용 가능한 모든 하위 명령 및 옵션에 대한 전체 설명은 ipadm(1M) 매뉴얼 페이지를 참조하십시오.

### IP 인터페이스 구성 제거(ipadm delete-ip)

데이터 링크를 통해 구성된 IP 인터페이스를 제거하려면 이 명령을 사용합니다. 이 명령은 특정 데이터 링크 구성을 수행할 때 특히 중요합니다. 예를 들어, 데이터 링크 이름 바꾸기는 IP 인터페이스가 해당 데이터 링크로 구성된 경우 실패합니다. 데이터 링크 이름을 바꾸기 전에 먼저 ipadm delete-ip를 실행해야 합니다.

일반적으로 이 명령은 시스템의 주 인터페이스 변경과 같이 다른 ipadm 및 dladm 하위 명령과 함께 사용됩니다. 이 작업을 위해서는 인터페이스를 삭제하고, 링크 이름을 바꾼 후, 이름이 바뀐 데이터 링크를 통해 인터페이스를 재구성해야 합니다. 순서는 다음과 같습니다.

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

주 인터페이스 변경에 대한 예제는 27 페이지 “데이터 링크 이름 바꾸기(dladm rename-link)”를 참조하십시오. 데이터 링크 이름이 바뀐 후 IP 인터페이스를 재구성하려면 18 페이지 “IP 인터페이스를 구성하는 방법”을 참조하십시오.

## IP 인터페이스 구성 사용 안함(ipadm disable-ip)

기본적으로 IP 인터페이스는 UP으로 플래그 지정되며 ipadm create-ip로 인터페이스를 만들 때 활성화 구성의 일부가 됩니다. ipadm disable-ip 하위 명령을 사용하면 해당 구성을 삭제하지 않아도 활성화 구성에서 인터페이스를 제거할 수 있습니다. 이 명령은 특정 인터페이스를 DOWN으로 플래그 지정합니다.

```
# ipadm disable-ip interface
```

IP 인터페이스를 작동하도록 만들고 해당 플래그를 UP으로 지정하려면 다음을 입력해야 합니다.

```
# ipadm enable-ip interface
```

---

**참고** - 인터페이스 상태를 표시하려면 ipadm을 사용합니다. 51 페이지 “IP 인터페이스에 대한 정보 가져오기”를 참조하십시오.

---

## 인터페이스의 주소 제거(ipadm delete-addr)

이 명령은 IP 인터페이스의 특정 주소 구성을 삭제합니다. 이 명령은 특정 인터페이스의 IP 주소를 변경하려고 할 때 유용합니다. 새 주소 구성을 지정하기 전 원래 주소 구성을 제거해야 합니다. 다음 일반 단계를 수행합니다.

```
# ipadm delete-addr addrobj
# ipadm create-addr parameters
```

인터페이스에 대한 IP 주소를 만드는 예제는 18 페이지 “IP 인터페이스를 구성하는 방법”을 참조하십시오.

주 - 하나의 인터페이스는 여러 주소를 가질 수 있습니다. 각 주소는 주소 객체로 식별됩니다. 올바른 주소를 제거하는지 확인하려면 주소 객체를 알아야 합니다. 시스템의 인터페이스 주소를 표시하려면 `ipadm show-addr` 하위 명령을 사용합니다. 주소 객체에 대한 설명은 18 페이지 “IP 인터페이스를 구성하는 방법”을 참조하십시오. 주소 표시에 대한 자세한 내용은 53 페이지 “IP 주소에 대한 정보 가져오기”를 참조하십시오.

## IP 인터페이스 등록 정보 설정

이 절에서는 `ipadm` 명령을 사용하여 선택한 IP 인터페이스 등록 정보를 설정하는 방법에 대해 설명합니다.

데이터 링크처럼 IP 인터페이스에도 특정 네트워크 환경에 맞게 사용자 정의할 수 있는 등록 정보가 있습니다. 각 인터페이스에 대해 IPv4 및 IPv6 프로토콜에 대해 하나씩 두 개의 등록 정보 세트가 존재합니다. MTU와 같은 일부 등록 정보는 데이터 링크와 IP 인터페이스에 공통됩니다. 따라서 데이터 링크의 MTU 값과 해당 링크에 구성된 인터페이스의 MTU 값을 다르게 지정할 수 있습니다. 해당 IP 인터페이스를 순회하는 IPv4 및 IPv6 패킷에 적용되는 다른 MTU 값을 지정할 수도 있습니다.

IP 인터페이스 등록 정보를 설정하는 데에는 세 가지 `ipadm` 하위 명령이 사용됩니다.

- `ipadm show-ifprop -p property interface` 하위 명령은 IP 인터페이스의 등록 정보 및 해당 현재 값을 표시합니다. `-p property` 옵션을 사용하지 않으면 IP 인터페이스의 모든 등록 정보가 나열됩니다. IP 인터페이스를 지정하지 않으면 모든 IP 인터페이스의 모든 등록 정보가 나열됩니다.
- `ipadm set-ifprop -p property =value interface` 하위 명령은 IP 인터페이스의 등록 정보에 값을 지정합니다.
- `ipadm reset-ifprop -p property interface` 하위 명령은 특정 등록 정보를 해당 기본값으로 재설정합니다.

## 패킷 전달 사용

네트워크에서 호스트는 다른 호스트 시스템으로 전송된 데이터 패킷을 받을 수 있습니다. 수신 로컬 시스템에서 패킷 전달을 사용으로 설정하면 해당 시스템이 데이터 패킷을 대상 호스트로 전달할 수 있습니다. 기본적으로 IP 전달은 사용 안함으로 설정됩니다.

패킷 전달은 IP 인터페이스 및 TCP/IP 프로토콜 모두에 설정할 수 있는 등록 정보로 관리됩니다. 패킷 전달 방식을 선택하려는 경우 IP 인터페이스에서 패킷 전달을 사용으로 설정합니다. 예를 들어, 여러 NIC가 포함된 시스템이 있을 수 있습니다. 일부 NIC는 외부 네트워크에 연결되고 다른 NIC는 개인 네트워크에 연결됩니다. 따라서 모든 인터페이스가 아니라 일부 인터페이스에서만 패킷 전달을 사용으로 설정합니다.

또한 TCP/IP 프로토콜의 등록 정보를 설정하여 시스템에서 패킷 전달을 전역으로 사용으로 설정할 수도 있습니다. 44 페이지 “전역적으로 패킷 전달 사용”을 참조하십시오.

주-IP 인터페이스 또는 프로토콜의 forwarding 등록 정보는 배타적이지 않습니다. 인터페이스 및 프로토콜에 대한 등록 정보를 동시에 설정할 수 있습니다. 예를 들어, 프로토콜에서 전역적으로 패킷 전달을 사용으로 설정한 후 시스템에서 각 IP 인터페이스에 대한 패킷 전달을 사용자 정의할 수 있습니다. 따라서 전역적으로 사용으로 설정되었더라도 시스템에 대해 패킷 전달을 선택적으로 적용할 수 있습니다.

IP 인터페이스에서 패킷 전달을 사용으로 설정하려면 다음 명령을 사용합니다.

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

여기서 *protocol-version*은 IPv4 또는 IPv6입니다. IPv4 및 IPv6 패킷에 대해 별도로 명령을 실행해야 합니다.

다음은 시스템에서 IPv4 패킷 전달만 사용으로 설정하는 방법에 대한 예제입니다.

```
# 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 주소 등록 정보 설정

ipadm 명령을 사용하면 이러한 주소를 인터페이스에 지정한 후 IP 주소 관련 등록 정보를 설정할 수 있습니다. 이러한 등록 정보를 설정하여 다음을 결정할 수 있습니다.

- 넷마스크 길이
- IP 주소를 아웃바운드 패킷의 소스 주소로 사용할 수 있는지 여부
- 주소가 전역 또는 비전역 영역에 속하는지 여부
- 주소가 개인 주소인지 여부

IP 주소 등록 정보 관련 작업을 수행할 때는 다음과 같은 ipadm 하위 명령을 사용합니다.

- ipadm show-addrprop [-p property] [addrobj] 하위 명령은 사용되는 옵션에 따라 주소 등록 정보를 표시합니다.

모든 IP 주소의 등록 정보를 나열하려면 등록 정보 또는 주소 객체를 지정하지 마십시오. 모든 IP 주소에 대해 단일 등록 정보의 값을 나열하려면 해당 등록 정보만 지정합니다. 특정 주소 객체의 모든 등록 정보를 나열하려면 해당 주소 객체만 지정합니다.

- `ipadm set-addrprop -p property=value addrobj` 하위 명령은 주소 등록 정보에 값을 지정합니다. 주소 등록 정보는 한 번에 하나만 설정할 수 있습니다.
- `ipadm reset-addrprop -p property addrobj` 하위 명령은 주소 등록 정보에 대한 모든 기본값을 복원합니다.

---

주 - 특정 인터페이스의 IP 주소를 변경하려면 `set-addressprop` 하위 명령을 사용하지 마십시오. 대신 주소 객체를 삭제하고 새 IP 주소를 사용하여 새 항목을 만듭니다. 40 페이지 “인터페이스의 주소 제거(`ipadm delete-addr`)”를 참조하십시오.

---

한 가지 예제로, IP 주소의 넷마스크를 변경한다고 가정해보십시오. IP 주소는 IP 인터페이스 `net3`에 구성되어 있으며 주소 객체 이름 `net3/v4`로 식별됩니다. 다음 명령은 넷마스크를 수정하는 방법을 보여줍니다.

```
# 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
```

## TCP/IP 프로토콜 등록 정보 설정

`ipadm` 명령을 사용하면 조정 가능 항목이라고도 하는 프로토콜 등록 정보를 구성할 수 있습니다. `ipadm`은 이전 릴리스에서 조정 가능 항목을 설정하는 데 주로 사용되었던 `ndd` 명령을 대체합니다.

TCP/IP 등록 정보는 인터페이스 기반이거나 전역일 수 있습니다. 특정 인터페이스나 영역의 모든 인터페이스에 전역적으로 등록 정보를 적용할 수 있습니다. 전역 등록 정보는 각 비전역 영역에서 다른 값을 사용할 수 있습니다. 지원되는 프로토콜 등록 정보 목록은 `ipadm(1M)` 매뉴얼 페이지를 참조하십시오.

일반적으로 TCP/IP 인터넷 프로토콜의 기본 값으로도 네트워크가 작동하는 데 충분합니다. 하지만 기본 값이 네트워크 토폴로지에 충분하지 않은 경우 필요에 따라 이러한 등록 정보를 사용자 정의할 수 있습니다.

TCP/IP 인터페이스 등록 정보를 설정하는 데에는 세 가지 `ipadm` 하위 명령이 사용됩니다.

- `ipadm show-prop -p property protocol` 명령은 프로토콜의 등록 정보 및 해당 현재 값을 표시합니다. `-p property` 옵션을 사용하지 않으면 프로토콜의 모든 등록 정보가 나열됩니다. 프로토콜을 지정하지 않으면 모든 프로토콜의 모든 등록 정보가 나열됩니다.
- `ipadm set-prop -p property =value protocol` 하위 명령은 IP 인터페이스의 등록 정보에 값을 지정합니다.
- `ipadm reset-prop -p property protocol` 하위 명령은 특정 프로토콜 등록 정보를 해당 기본값으로 재설정합니다.

주 - 등록 정보가 여러 값을 수신할 수 있는 경우 다음과 같이 += 수식자를 사용하여 등록 정보에 여러 값을 지정합니다.

```
ipadm set-prop -p property+=value1 [value2 value3 ...].
```

등록 정보의 값 세트에서 값을 하나 제거하려면 다음과 같이 -= 수식자를 사용합니다.

```
ipadm set-prop -p property-=value2
```

## 전역적으로 패킷 전달 사용

41 페이지 “패킷 전달 사용”에서는 인터페이스에서 패킷 전달을 사용으로 설정하는 방법을 보여줍니다. IP 인터페이스 등록 정보에 패킷 전달을 설정하면 이 기능을 선택적으로 구현할 수 있습니다. 시스템의 특정 인터페이스에서만 이 등록 정보를 사용으로 설정할 수 있습니다.

IP 인터페이스 수에 관계없이 전체 시스템에서 패킷 전달을 사용으로 설정하려는 경우, 프로토콜 등록 정보를 사용합니다. 프로토콜에서 등록 정보 이름은 IP 인터페이스와 동일한 forwarding입니다. IPv4 및 IPv6 프로토콜에서 패킷 전달을 사용으로 설정하려면 명령을 별도로 실행해야 합니다.

다음 예제에서는 시스템에서 모든 IPv4 및 IPv6 트래픽에 대해 패킷 전달을 사용으로 설정하는 방법을 보여줍니다.

```
# ipadm show-prop -p forwarding ip
PROTO  PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
ipv4   forwarding rw    off      --          off      on,off
ipv6   forwarding rw    off      --          off      on,off
#
# ipadm set-prop -p forwarding=on ipv4
# ipadm set-prop -p forwarding=on ipv6
#
# ipadm show-prop ip
PROTO  PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
ipv4   forwarding rw    on       on         off      on,off
```

ipv4	ttl	rw	255	--	255	1-255
ipv6	forwarding	rw	on	on	off	on,off
ipv6	hoplimit	rw	255	--	255	1-255#

주 - IP 인터페이스 또는 프로토콜의 forwarding 등록 정보는 배타적이지 않습니다. 인터페이스 및 프로토콜에 대한 등록 정보를 동시에 설정할 수 있습니다. 예를 들어, 프로토콜에서 전역적으로 패킷 전달을 사용으로 설정한 후 시스템에서 각 IP 인터페이스에 대한 패킷 전달을 사용자 정의할 수 있습니다. 따라서 전역적으로 사용으로 설정되었더라도 시스템에 대해 패킷 전달을 선택적으로 적용할 수 있습니다.

## 권한이 부여된 포트 설정

TCP, UDP 및 SCTP와 같은 전송 프로토콜에서 1-1023 포트는 root 권한으로 실행되는 프로세스만 이러한 포트에 바인딩할 수 있는 권한이 부여된 기본 포트입니다. ipadm 명령을 사용하면 권한이 부여된 포트가 되도록 이 지정된 기본 범위를 초과하여 포트를 예약할 수 있습니다. 따라서 root 프로세스만 해당 포트에 바인딩할 수 있습니다. 권한이 부여된 포트를 설정하려면 다음과 같은 전송 프로토콜 등록 정보를 사용자 정의합니다.

- `smallest_nonpriv_port` - 해당 값이 일반 사용자가 바인딩할 수 있는 포트 번호 범위를 나타내는 등록 정보입니다. 지정된 포트가 이 범위 내에 있으면 권한이 부여된 포트에 설정할 수 있습니다. 등록 정보 값을 표시하려면 `ipadm show-prop` 명령을 사용합니다.
- `extra_priv_ports` - 권한이 부여된 포트를 지정하는 등록 정보입니다. 제한할 포트를 지정하려면 `ipadm set-prop` 하위 명령을 사용합니다. 이 등록 정보에는 여러 값을 지정할 수 있습니다.

예를 들어, TCP 포트 3001 및 3050을 root 사용자로만 액세스가 제한된 권한이 부여된 포트에 설정한다고 가정해보십시오. `smallest_nonpriv_port` 등록 정보는 1024가 권한이 부여되지 않은 포트에 대한 최저 포트 번호임을 나타냅니다. 따라서 지정된 포트 3001 및 3050을 권한이 부여된 포트에 변경할 수 있습니다. 이렇게 하려면 다음과 유사한 명령을 실행합니다.

```
# ipadm show-prop -p smallest_nonpriv_port tcp
PROTO PROPERTY          PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
tcp  smallest_nonpriv_port  rw    1024    --          1024    1024-32768

# ipadm show-prop -p extra_priv_ports tcp
PROTO  PROPERTY          PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
tcp    extra_priv_ports  rw    2049,4045  --          2049,4045  1-65535

# ipadm set-prop -p extra_priv_ports+=3001 tcp
# ipadm set-prop -p extra_priv_ports+=3050 tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO  PROPERTY          PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE
tcp    extra_priv_ports  rw    2049,4045  3001,3050  2049,4045  1-65535
                                     3001,3050
```

권한이 부여된 포트에서 포트 중 하나(예: 4045)를 제거하려면 다음 명령을 입력합니다.

```
# ipadm set-prop -p extra_priv_ports=4045 tcp
# ipadm show-prop -p extra_priv_ports tcp
PROTO PROPERTY PERM CURRENT PERSISTENT DEFAULT POSSIBLE
tcp extra_priv_ports rw 2049,3001 3001,3050 2049,4045 1-65535
3050
```

## 멀티홉 호스트에 대칭 경로 지정 구현

기본적으로 여러 인터페이스가 있는 시스템은 **멀티홉 호스트**라고도 하며, 경로 지정 테이블에서 트래픽 대상까지 가장 긴 일치 경로를 기준으로 네트워크 트래픽의 경로를 지정합니다. 대상까지의 길이가 같은 경로가 여러 개 있을 경우 Oracle Solaris는 ECMP(Equal-Cost Multi-Path) 알고리즘을 적용하여 트래픽을 해당 경로에 분산시킵니다.

이런 방식의 트래픽 분산이 적합하지 않은 경우도 있습니다. 패킷의 IP 소스 주소와 동일한 서브넷에 없는 멀티홉 호스트의 인터페이스를 통해 IP 패킷이 전송될 수 있습니다. 또한 송신 패킷이 특정 수신 요청(예: ICMP 에코 요청)에 대한 응답인 경우 요청과 응답이 동일한 인터페이스를 순회할 수 없습니다. 이러한 트래픽 경로 지정 구성을 **비대칭 경로 지정**이라고 합니다. 인터넷 서비스 공급자가 RFC 3704(<http://rfc-editor.org/rfc/bcp/bcp84.txt>)에 설명된 대로 진입 필터링을 구현하는 경우 비대칭 경로 지정 구성으로 인해 공급자가 송신 패킷을 삭제할 수도 있습니다.

RFC 3704는 인터넷에서 서비스 거부 공격을 제한하기 위한 것입니다. 이 의도를 준수하려면 네트워크에서 대칭 경로 지정을 구성해야 합니다. Oracle Solaris에서 IP hostmodel 등록 정보를 사용하면 이 요구 사항을 충족할 수 있습니다. 이 등록 정보는 멀티홉 호스트를 통해 수신 또는 전송된 IP 패킷의 동작을 제어합니다.

hostmodel 등록 정보는 세 가지 가능한 값 중 하나를 가질 수 있습니다.

- strong** RFC 1122에 정의된 강력한 ES(엔드 시스템) 모델에 해당합니다. 이 값은 대칭 경로 지정을 구현합니다.
- weak** RFC 1122에 정의된 약한 ES 모델에 해당합니다. 이 값에서는 멀티홉 호스트가 비대칭 경로 지정을 사용합니다.
- src-priority** 기본 경로를 사용하여 패킷 경로 지정을 구성합니다. 경로 지정 테이블에 대상 경로가 여러 개 있는 경우 기본 경로는 송신 패킷의 IP 소스 주소가 구성된 인터페이스를 사용하는 경로입니다. 해당 경로가 없는 경우 송신 패킷은 패킷의 IP 대상에 대한 가장 긴 일치 경로를 사용합니다.

다음 예제에서는 멀티홉 호스트에서 IP 패킷의 대칭 경로 지정을 구현하는 방법을 보여줍니다.

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

```

src-priority,
weak
ip4 hostmodel rw strong -- weak strong,
src-priority,
weak

```

## 트래픽 혼잡 제어 구현

네트워크 혼잡은 네트워크가 수용할 수 있는 것보다 많은 수의 패킷을 노드가 전송할 때 일반적으로 라우터 버퍼 오버플로우의 형태로 발생합니다. 여러 알고리즘에서 전송 시스템에 대한 제어를 설정하여 트래픽 혼잡을 방지합니다. 이러한 알고리즘은 Oracle Solaris에서 지원되며 운영 체제에 쉽게 추가하거나 직접 플러그인할 수 있습니다.

다음 표에서는 지원되는 알고리즘을 나열하고 설명합니다.

알고리즘	Oracle Solaris 이름	설명
NewReno	newreno	Oracle Solaris의 기본 알고리즘입니다. 제어 방식에는 발신자의 혼잡 윈도우, 느린 시작 및 혼잡 회피가 포함됩니다.
HighSpeed	highspeed	고속 네트워크를 위한 가장 잘 알려지고 가장 단순한 NewReno 수정 버전 중 하나입니다.
CUBIC	cubic	Linux 2.6에서 현재 기본 알고리즘입니다. 혼잡 회피 위상을 선형 윈도우 증가에서 3차 함수로 변경합니다.
Vegas	vegas	실제 패킷 손실을 트리거하지 않고 혼잡을 예측하려고 시도하는 고전적인 지연 기반 알고리즘입니다.

혼잡 제어는 다음과 같은 제어 관련 TCP 등록 정보를 설정하여 사용으로 설정됩니다. 이러한 등록 정보는 TCP용으로 나열되지만 해당 등록 정보로 사용으로 설정되는 제어 방식은 SCTP 트래픽에도 적용됩니다.

- **cong\_enabled** - 현재 시스템에서 운영되는 콤마로 구분된 알고리즘 목록을 포함합니다. 알고리즘을 추가 또는 제거하여 사용하려는 알고리즘만 사용으로 설정할 수 있습니다. 이 등록 정보는 여러 값을 포함할 수 있습니다. 따라서 적용하려는 변경 사항에 따라 += 수식자 또는 -= 수식자를 사용해야 합니다.
- **cong\_default** - 응용 프로그램에서 소켓 옵션에 명시적으로 알고리즘을 지정하지 않을 때 기본적으로 사용되는 알고리즘입니다. 현재까지 **cong\_default** 등록 정보 값은 전역 및 비전역 영역에 모두 적용됩니다.

프로토콜에 혼잡 제어를 위한 알고리즘을 추가하려면 다음 명령을 실행합니다.

```
# ipadm set-prop -p cong_enabled+=algorithm tcp
```

알고리즘을 제거하려면 다음 명령을 실행합니다.

```
# ipadm set-prop -p cong_enabled-=algorithm tcp
```

기본 알고리즘을 바꾸려면 다음 명령을 실행합니다.

```
# ipadm set-prop -p cong_default=algorithm tcp
```

---

주 - 알고리즘을 추가 또는 제거할 때 따라야 할 시퀀스 규칙은 없습니다. 알고리즘을 등록 정보에 추가하기 전 알고리즘을 제거할 수 있습니다. 하지만 `cong_default` 등록 정보는 항상 알고리즘이 정의되어 있어야 합니다.

---

다음 예제에서는 혼잡 제어를 구현하기 위해 수행할 수 있는 단계를 보여줍니다. 이 예제에서 TCP 프로토콜의 기본 알고리즘은 `newreno`에서 `cubic`으로 변경됩니다. 그런 후 `vegas` 알고리즘이 사용으로 설정된 알고리즘 목록에서 제거됩니다.

```
# ipadm show-prop -p cong_default,cong_enabled tcp
```

PROTO	PROPERTY	PERM	CURRENT	PERSISTENT	DEFAULT	POSSIBLE
tcp	cong_default	rw	newreno	--	newreno	-
tcp	cong_enabled	rw	newreno,cubic, highspeed, vegas	--	newreno	newreno,cubic, highspeed,vegas

```
# ipadm set-prop -p cong_enabled-=vegas tcp
# ipadm set-prop -p cong_default=cubic tcp
```

```
# ipadm show-prop -p cong_default,cong_enabled tcp
```

PROTO	PROPERTY	PERM	CURRENT	PERSISTENT	DEFAULT	POSSIBLE
tcp	cong_default	rw	cubic	--	newreno	-
tcp	cong_enabled	rw	newreno,cubic, highspeed	--	newreno	newreno,cubic, highspeed,vegas

## TCP 수신 버퍼 크기 변경

TCP 수신 버퍼 크기는 TCP 등록 정보 `recv_buf`(기본값 128KB)를 사용하여 설정됩니다. 하지만 응용 프로그램은 사용 가능한 대역폭을 동일하게 사용하지 않습니다. 따라서 연결 대기 시간에 따라 기본 크기를 변경해야 할 수 있습니다. 예를 들어, Oracle Solaris의 보안 셸 기능을 사용하면 데이터 스트림에서 수행되는 추가 체크섬 및 암호화 프로세스로 인해 대역폭 사용에 대한 오버헤드가 발생합니다. 따라서 버퍼 크기를 늘려야 할 수 있습니다. 마찬가지로, 대량 전송 보류를 수행하는 응용 프로그램이 대역폭을 효율적으로 사용하도록 하려면 동일한 버퍼 크기 조정이 필요합니다.

다음과 같이 *BDP(대역폭 지연 곱)*를 예상하여 사용할 올바른 수신 버퍼 크기를 계산할 수 있습니다.

$$BDP = available\_bandwidth * connection\_latency$$

연결 대기 시간 값을 가져오려면 `ping -s host`를 사용합니다. 대역폭 사용을 예상하려면 `iperf` 및 `iperf` 도구를 사용합니다.

적합한 수신 버퍼 크기는 BDP 값에 가깝습니다. 하지만 대역폭 사용은 다양한 옵션에 따라 달라질 수 있습니다. 공유 기반구조 또는 응용 프로그램 수 및 대역폭 사용을 경험하는 사용자로 인해 예상값이 변경될 수 있습니다.

버퍼 크기 값을 변경하려면 다음 구문을 사용하십시오.

```
# ipadm set-prop -p recv_buf=value tcp
```

다음 예제에서는 버퍼 크기를 164KB로 늘리는 방법을 보여줍니다.

```
# ipadm show-prop -p recv_buf tcp
PROTO PROPERTY  PERM CURRENT  PERSISTENT  DEFAULT  POSSIBLE
tcp  recv_buf  rw  128000  --  128000  2048-1048576
```

```
# ipadm set-prop -p recv_buf=164000 tcp
```

```
# ipadm show-prop -p recv_buf tcp
PROTO PROPERTY  PERM CURRENT  PERSISTENT  DEFAULT  POSSIBLE
tcp  recv_buf  rw  164000  --  164000  2048-1048576
```

선호 크기는 각 상황에 따라 달라지므로 버퍼 크기에 대해 선호되는 설정 값은 특정되지 않습니다. 다음은 고유한 특정 조건에서 각 네트워크에 대해 서로 다른 BDP 값을 보여주는 사례입니다.

버퍼 크기의 기본값이 128KB인 일반적인 1Gbps LAN:

$$\text{BDP} = 128 \text{ MBps} * 0.001 \text{ s} = 128 \text{ kB}$$

대기 시간이 100ms인 이론적인 1Gbps WAN:

$$\text{BDP} = 128 \text{ MBps} * 0.1 \text{ s} = 12.8 \text{ MB}$$

유럽-미국 링크(iperf로 측정된 대역폭)

$$\text{BDP} = 2.6 \text{ MBps} * 0.175 = 470 \text{ kB}$$

BDP를 계산할 수 없으면 다음과 같은 관측값을 지침으로 사용하십시오.

- LAN 기반 대량 전송의 경우 버퍼 크기의 기본값인 128KB로도 충분합니다.
- 대부분의 WAN 배포에서 수신 버퍼 크기는 2MB 범위 내에 있어야 합니다.



**주의** - TCP 수신 버퍼 크기를 늘리면 여러 네트워크 응용 프로그램의 메모리 사용 공간이 늘어납니다.

## IP 인터페이스 및 주소 모니터링

ipadm 명령을 사용하면 IP 인터페이스 및 해당 등록 정보를 모니터하고 정보를 가져올 수 있습니다. 이 명령을 단독으로 사용하면 시스템에서 IP 인터페이스에 대한 일반 정보가 표시됩니다. 하지만 다음 구문을 사용해서 표시하려는 정보를 제한하기 위한 하위 명령을 사용할 수도 있습니다.

**ipadm show-\*** [*other-arguments*] [*interface*]

- 인터페이스 정보만 가져오려면 ipadm show-if를 사용합니다.
- 주소 정보만 가져오려면 ipadm show-addr를 사용합니다.
- 인터페이스 등록 정보에 대한 정보를 가져오려면 ipadm show-ifprop를 사용합니다.
- 주소 등록 정보에 대한 정보를 가져오려면 ipadm show-addrprop를 사용합니다.

이 절에서는 인터페이스 정보를 가져오기 위해 ipadm 하위 명령을 사용하는 방법에 대한 몇 가지 예제를 제공합니다. ipadm show-\* 명령으로 표시되는 모든 필드에 대한 자세한 내용은 ipadm(1M) 매뉴얼 페이지를 참조하십시오.

## IP 인터페이스에 대한 일반 정보 가져오기

하위 명령을 사용하지 않고 ipadm 명령을 사용하면 모든 시스템의 IP 인터페이스에 대한 기본 정보를 제공합니다. 예를 들면 다음과 같습니다.

```
# 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
```

샘플 출력 결과에서는 다음 정보를 제공합니다.

- IP 인터페이스
- 각 인터페이스의 클래스
- 각 인터페이스의 상태

- “독립형” IP 인터페이스 또는 다른 유형의 인터페이스 구성을 위한 기본 인터페이스가 되는 인터페이스 상태. 예제에서 net1 및 net2는 UNDER 열에 표시된 것처럼 `imp0`의 기본 인터페이스입니다.
- 인터페이스와 연관된 주소 객체. 주소 객체는 특정 IP 주소를 식별합니다. 이러한 주소 객체는 인터페이스 이름과 구분하기 위해 NAME 제목 아래에 나열되고 들여써집니다.
- CLASS/TYPE 제목 아래에 들여써지고 `static`, `dhcp` 등일 수 있는 IP 주소 유형
- ADDRESS 열 아래에 나열된 실제 주소

따라서 `ipadm` 명령은 시스템의 인터페이스에 대한 포괄적인 개요를 제공합니다.

## IP 인터페이스에 대한 정보 가져오기

IP 인터페이스에 대한 자세한 내용을 보려면 `ipadm show-if [interface]` 하위 명령을 사용하십시오. 인터페이스를 지정하지 않으면 시스템의 모든 인터페이스에 대한 정보가 제공됩니다.

명령 출력 결과의 필드는 다음을 나타냅니다.

**IFNAME** 정보가 표시되는 인터페이스를 나타냅니다.

**CLASS** 다음 네 가지 중 하나일 수 있는 인터페이스 클래스를 나타냅니다.

- `ip`는 IP 인터페이스를 나타냅니다.
- `imp`는 IPMP 인터페이스를 나타냅니다.
- `vni`는 가상 인터페이스를 나타냅니다.
- `loopback`은 자동으로 생성되는 루프백 인터페이스를 나타냅니다. 루프백 인터페이스를 제외하고 나머지 세 인터페이스 클래스는 수동으로 만들 수 있습니다.

**STATE** `ok`, `offline`, `failed`, `down` 또는 `disabled`일 수 있는 인터페이스 상태를 나타냅니다.

`failed` 상태는 IPMP 그룹에 적용되며 작동 중지되고 트래픽을 호스트할 수 없는 데이터 링크 또는 IP 인터페이스를 나타낼 수 있습니다. IP 인터페이스가 IPMP 그룹에 속하는 경우 IPMP 인터페이스가 그룹의 다른 활성 IP 인터페이스를 사용하여 계속 트래픽을 받고 보낼 수 있습니다.

`down` 상태는 관리자가 오프라인 상태로 전환한 IP 인터페이스를 나타냅니다.

`disable` 상태는 `ipadm disable-if` 명령을 사용하여 연결 취소된 IP 인터페이스를 나타냅니다.

**ACTIVE** 인터페이스가 트래픽을 호스트하는 데 사용되는지 여부를 나타내며 `yes` 또는 `no`로 설정됩니다.

OVER 인터페이스의 IPMP 클래스에만 적용되며 IPMP 인터페이스 또는 그룹을 구성하는 기본 인터페이스를 나타냅니다.

다음은 이 명령이 제공하는 정보에 대한 예제입니다.

```
# 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 인터페이스 등록 정보에 대한 정보가져오기

IP 인터페이스의 등록 정보에 대한 자세한 내용을 보려면 `ipadm show-ifprop [interface]` 명령을 사용하십시오. 등록 정보 또는 인터페이스를 지정하지 않으면 시스템에서 모든 IP 인터페이스의 모든 등록 정보에 대한 정보가 제공됩니다.

명령 출력 결과의 필드는 다음을 나타냅니다.

- IFNAME 정보가 표시되는 IP 인터페이스를 나타냅니다.
- PROPERTY 인터페이스의 등록 정보를 나타냅니다. 한 인터페이스에 등록 정보가 여러 개 있을 수 있습니다.
- PROTO 등록 정보가 적용되고 IPv4 또는 IPv6일 수 있는 프로토콜을 나타냅니다.
- PERM 지정된 등록 정보에 대해 허용된 권한을 나타내며 읽기 전용, 쓰기 전용 또는 둘 다일 수 있습니다.
- CURRENT 활성 구성에서 등록 정보의 현재 값을 나타냅니다.
- PERSISTENT 시스템을 재부트할 때 재적용되는 등록 정보의 값을 나타냅니다.
- DEFAULT 지정한 등록 정보의 기본값을 나타냅니다.
- POSSIBLE 지정한 등록 정보에 지정될 수 있는 값의 목록을 나타냅니다. 숫자 값의 경우 허용되는 값의 범위가 표시됩니다.

주- 정보가 요청되는 등록 정보를 인터페이스가 지원하지 않는 경우와 같이 필드 값을 알 수 없는 경우 값이 물음표(?)로 표시됩니다.

다음은 `ipadm show-ifprop` 하위 명령이 제공하는 정보에 대한 예제입니다.

```
# 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 주소에 대한 정보 가져오기

IP 주소에 대한 자세한 내용을 보려면 `ipadm show-addr [interface]` 하위 명령을 사용하십시오. 인터페이스를 지정하지 않으면 시스템에서 모든 IP 주소에 대한 정보가 제공됩니다.

명령 출력 결과의 필드는 다음을 나타냅니다.

ADDROBJ	IP 주소가 나열되는 주소 객체를 지정합니다.
TYPE	IP 주소가 <code>static</code> , <code>dhcp</code> 또는 <code>addrconf</code> 인지를 나타냅니다. <code>addrconf</code> 값은 <code>Stateless</code> 또는 <code>Stateful</code> 주소 구성을 사용하여 주소를 가져왔음을 나타냅니다.
STATE	활성 구성에서 주소 객체의 상태에 대해 설명합니다. 이러한 값의 전체 목록은 <code>ipadm(1M)</code> 매뉴얼 페이지를 참조하십시오.
ADDR	인터페이스에 구성되는 IP 주소를 지정합니다. 주소는 IPv4 또는 IPv6일 수 있습니다. 터널 인터페이스는 로컬 및 원격 주소를 모두 표시합니다.

터널에 대한 자세한 내용은 [Oracle Solaris 11.1 네트워크 구성 및 관리의 6 장, “IP 터널 구성”](#)을 참조하십시오.

다음은 `ipadm show-addr` 하위 명령이 제공하는 정보에 대한 예제입니다.

```
# 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
```

이 명령에 인터페이스를 지정하고 인터페이스에 여러 주소가 포함된 경우 다음과 비슷한 정보가 표시됩니다.

```
# 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/?`로 나열된 주소 객체는 `libipadm` API를 사용하지 않는 응용 프로그램이 인터페이스에 주소를 구성했음을 나타냅니다. 이러한 응용 프로그램은 `ipadm` 명령의 제어를 받지 않습니다. 이 명령을 사용하려면 주소 객체 이름이 `interface/user-defined-string` 형식을 사용해야 합니다. IP 주소 지정의 예는 [18 페이지 “IP 인터페이스를 구성하는 방법”](#)을 참조하십시오.

## IP 주소 등록 정보에 대한 정보 가져오기

IP 주소 등록 정보에 대한 자세한 내용을 보려면 `ipadm show-addrprop [addrobj]` 하위 명령을 사용하십시오. 모든 등록 정보를 나열하려면 `addrobj` 옵션을 생략합니다. 모든 IP 주소에 대한 단일 등록 정보를 나열하려면 해당 등록 정보만 지정합니다. 특정 주소의 모든 등록 정보를 나열하려면 `addrobj` 옵션만 지정합니다.

명령 출력 결과의 필드는 다음을 나타냅니다.

ADDROBJ	등록 정보가 나열되는 주소 객체를 나타냅니다.
PROPERTY	주소 객체의 등록 정보를 나타냅니다. 한 주소 객체에 등록 정보가 여러 개 있을 수 있습니다.
PERM	지정된 등록 정보에 대해 허용된 권한을 나타내며 읽기 전용, 쓰기 전용 또는 둘 다일 수 있습니다.
CURRENT	현재 구성에서 등록 정보의 실제 값을 나타냅니다.
PERSISTENT	시스템을 재부트할 때 재적용되는 등록 정보의 값을 나타냅니다.
DEFAULT	지정한 등록 정보의 기본값을 나타냅니다.
POSSIBLE	지정한 등록 정보에 지정될 수 있는 값의 목록을 나타냅니다. 숫자 값의 경우 허용되는 값의 범위가 표시됩니다.

다음은 `ipadm show-addrprop` 하위 명령이 제공하는 정보에 대한 예제입니다.

```
# 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          --
```

# Oracle Solaris 실행 랩탑에서 무선 네트워킹 구성

IEEE 802.11 사양은 근거리 통신망에 대한 무선 통신을 정의합니다. 이러한 사양과 사양에서 설명하는 네트워크를 총체적으로 *WiFi*라고 합니다. 이 용어는 Wi-Fi Alliance 무역 그룹의 상표입니다. WiFi 네트워크는 공급자와 잠재 클라이언트가 모두 쉽게 구성할 수 있습니다. 따라서 점차 인기가 증가하고 있으며 전세계에서 일반적으로 사용되고 있습니다. WiFi 네트워크는 휴대폰, TV 및 라디오와 동일한 전파 기술을 사용합니다.

주 - WiFi 서버 또는 액세스 포인트를 구성하는 기능은 Oracle Solaris에 포함되어 있지 않습니다.

다음 항목을 다룹니다.

- 55 페이지 “WiFi 통신 작업 맵”
- 61 페이지 “보안 WiFi 통신”

## WiFi 통신 작업 맵

작업	설명	수행 방법
WiFi 네트워크에 연결합니다.	로컬 WiFi 네트워크를 설치하고 통신을 설정합니다.	56 페이지 “WiFi 네트워크에 연결하는 방법”
WiFi 링크에서 통신을 모니터링합니다.	표준 Oracle Solaris 네트워킹 도구를 사용하여 WiFi 링크 상태를 확인합니다.	59 페이지 “WiFi 링크를 모니터링하는 방법”
보안 WiFi 통신을 설정합니다.	WEP(Wired Equivalent Privacy) 키를 만들고 이 키를 사용하여 보안 WiFi 네트워크와 연결을 설정합니다.	61 페이지 “암호화된 WiFi 네트워크 연결을 설정하는 방법”

## ▼ WiFi 네트워크에 연결하는 방법

시작하기 전에 다음 단계를 수행하여 랩탑을 WiFi 네트워크에 연결합니다.

### 1 관리자로 전환합니다.

자세한 내용은 **Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”**을 참조하십시오.

### 2 데이터 링크의 물리적 속성을 표시합니다.

```
# dladm show-phys
LINK          MEDIA          STATE    SPEED  DUPLEX    DEVICE
net0          Ethernet      up       1500   full     ath0
net1          Ethernet      up       1500   full     e1000g0
```

이 출력 결과 예에는 두 개의 링크를 사용할 수 있음을 나타냅니다. 장치 `ath0` 링크를 통해 `Net0`은 WiFi 통신을 지원합니다. `e1000g0` 링크는 시스템을 유선 네트워크에 연결하는 역할을 수행합니다.

### 3 WiFi 인터페이스를 구성합니다.

다음 단계를 사용하여 인터페이스를 구성합니다.

#### a. WiFi를 지원하는 인터페이스를 만듭니다.

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

#### b. 링크가 연결(plumb)되었는지 확인합니다.

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

### 4 사용 가능한 네트워크를 확인합니다.

```
# 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` 명령의 출력 결과 예에는 현재 위치에서 사용 가능한 WiFi 네트워크에 대한 정보가 표시됩니다. 출력 결과에 포함되는 정보는 다음과 같습니다.

**LINK** WiFi 연결에서 사용할 링크 이름을 나타냅니다.

**ESSID** Extended Service Set ID를 나타냅니다. ESSID는 특정 무선 네트워크의 관리자가 임의로 이름을 지정할 수 있는 WiFi 네트워크의 이름입니다.

**BSSID/IBSSID** 특정 ESSID의 고유 식별자인 Basic Service Set ID를 나타냅니다. BSSID는 네트워크에 특정 ESSID를 제공하는 주변 액세스 포인트의 48비트 MAC 주소입니다.

SEC	네트워크 액세스에 필요한 보안 유형을 나타냅니다. 값은 none 또는 WEP입니다. WEP에 대한 자세한 내용은 61 페이지 “보안 WiFi 통신”을 참조하십시오.
STRENGTH	해당 위치에서 사용할 수 있는 WiFi 네트워크의 라디오 신호 강도를 나타냅니다.
MODE	네트워크에서 실행하는 802.11 프로토콜의 버전을 나타냅니다. 모드는 a, b, g 또는 이러한 모드의 조합입니다.
SPEED	특정 네트워크의 초당 메가비트 속도를 나타냅니다.

## 5 WiFi 네트워크에 연결합니다.

다음 중 하나를 수행합니다.

- 신호가 가장 강한 비보안 WiFi 네트워크에 연결합니다.

```
# dladm connect-wifi
```

- ESSID를 지정하여 비보안 네트워크에 연결합니다.

```
# dladm connect-wifi -e ESSID
```

dladm의 connect-wifi 하위 명령에는 WiFi 네트워크에 연결하는 추가 옵션이 여러 개 있습니다. 자세한 내용은 [dladm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

## 6 인터페이스의 IP 주소를 구성합니다.

다음 중 하나를 수행합니다.

- DHCP 서버에서 IP 주소를 받습니다.

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

WiFi 네트워크가 DHCP를 지원하지 않는 경우 다음 메시지가 수신됩니다.

```
ipadm: interface: interface does not exist or cannot be managed using DHCP
```

- 정적 IP 주소를 구성합니다.

시스템에 전용 IP 주소가 있는 경우 이 옵션을 사용합니다.

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

## 7 시스템이 연결된 WiFi 네트워크의 상태를 확인합니다.

```
# dladm show-wifi
```

LINK	STATUS	ESSID	SEC	STRENGTH	MODE	SPEED
net0	connected	ofc	none	very good	g	36Mb

이 출력 결과 예는 시스템이 현재 ofc 네트워크에 연결되어 있음을 나타냅니다. 4단계의 이전 scan-wifi 출력에는 사용 가능한 네트워크 중 ofc의 신호가 가장 강한 것으로 표시되었습니다. 다른 네트워크를 직접 지정하지 않는 경우 dladm connect-wifi 명령은 신호가 가장 강한 WiFi 네트워크를 자동으로 선택합니다.

**8 WiFi 네트워크를 통해 인터넷에 액세스합니다.**

시스템이 연결되어 있는 네트워크에 따라 다음 중 하나를 수행합니다.

- 액세스 포인트가 무료 서비스를 제공하는 경우 이제 선택한 브라우저나 응용 프로그램을 실행할 수 있습니다.
- 액세스 포인트가 요금을 지불해야 하는 상용 WiFi 네트워크에 있는 경우 현재 위치에서 제공되는 지침을 따릅니다. 일반적으로 브라우저를 실행하고 키를 제공하고 네트워크 공급자에게 신용 카드 정보를 제공합니다.

**9 세션을 종료합니다.**

다음 중 하나를 수행합니다.

- WiFi 세션을 종료하지만 시스템이 계속 실행되도록 합니다.

```
# dladm disconnect-wifi
```

- 현재 여러 세션이 실행되고 있는 경우 특정 WiFi 세션을 종료합니다.

```
# dladm disconnect-wifi link
```

여기서 *link*는 세션에 사용 중인 인터페이스를 나타냅니다.

- WiFi 세션이 실행되는 동안 시스템을 정상적으로 종료합니다.

```
# shutdown -g0 -i5
```

`shutdown` 명령을 통해 시스템을 끄기 전에 WiFi 세션의 연결을 명시적으로 끊지 않아도 됩니다.

**예 5-1 특정 WiFi 네트워크에 연결**

다음 예제에서는 Oracle Solaris 랩탑을 무선 네트워크에 연결하기 위해 수행하는 서로 다른 단계를 조합해서 보여줍니다. 이 예제에서는 또한 OS가 무선 네트워크를 임의로 선택하도록 허용하는 대신 시스템이 특정 신호 무선 네트워크에 강제로 연결하도록 지정하는 방법을 보여줍니다. 이 예제에서는 정적 IP 주소 `10.192.16.3/24`가 랩탑에 구성되어 있다고 가정합니다. 이 예제는 먼저 WiFi 링크의 가용성을 확인합니다.

```
# 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
```

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

# 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

# dladm show-wifi
LINK        STATUS      ESSID      SEC      STRENGTH    MODE    SPEED
net0        connected   citinet    none     good        g       11Mb

```

브라우저나 다른 응용 프로그램을 실행하여 WiFi 네트워크를 통해 작업을 시작합니다.

```
# firefox
```

Firefox 브라우저의 홈 페이지가 나타납니다.

세션을 종료하지만 랩탑이 계속 실행되도록 둡니다.

```

# dladm disconnect-wifi
# dladm show-wifi
LINK        STATUS      ESSID      SEC      STRENGTH    MODE    SPEED
net0        disconnected --          --        --        --        --

```

show-wifi 출력 결과에서 net0 링크가 WiFi 네트워크에서 연결이 끊어졌음을 확인합니다.

## ▼ WiFi 링크를 모니터하는 방법

이 절차에서는 표준 네트워킹 도구를 통해 WiFi 링크 상태를 모니터하고 linkprop 하위 명령을 통해 선택된 링크 등록 정보를 변경하는 방법을 보여줍니다.

### 1 관리자로 전환합니다.

자세한 내용은 [Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”](#)을 참조하십시오.

### 2 56 페이지 “WiFi 네트워크에 연결하는 방법”에 설명된 대로 WiFi 네트워크에 연결합니다.

### 3 링크의 등록 정보를 봅니다.

다음 구문을 사용하십시오.

```
# dladm show-linkprop link
```

예를 들어, 다음 구문을 사용하여 net0 무선 링크에 설정된 연결의 상태를 표시합니다.

```
# dladm show-linkprop net0
...
PROPERTY          VALUE          DEFAULT        POSSIBLE
channel            5              --             --
powermode          off            off            off,fast,max
radio              ?              on             on,off
speed              36            --             1,2,5.5,6,9,11,12,18,24,36,48,54
...
```

#### 4 링크에 고정 속도를 설정합니다.



주의 - Oracle Solaris는 WiFi 연결의 최적 속도를 자동으로 선택합니다. 링크의 초기 속도를 수정하면 성능이 감소되거나 특정 WiFi 연결이 설정되지 않을 수 있습니다.

show-linkprop 출력 결과에 나열된 가능한 속도 값 중 하나로 링크 속도를 수정할 수 있습니다.

```
# dladm set-linkprop -p speed=value link
```

#### 5 링크를 통해 패킷 플로우를 확인합니다.

```
# netstat -I net0 -i 5
      input  net0      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
```

### 예 5-2 링크 속도 설정

이 예에서는 WiFi 네트워크에 연결한 후 링크 속도를 설정하는 방법을 보여줍니다.

```
# dladm show-linkprop -p speed net0
PROPERTY          VALUE          DEFAULT        POSSIBLE
speed              24            --             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
PROPERTY          VALUE          DEFAULT        POSSIBLE
speed              36            --             1,2,5,6,9,11,12,18,24,36,48,54
```

## 보안 WiFi 통신

전과 기술을 통해 WiFi 네트워크를 쉽게 사용할 수 있으며 여러 위치에서 사용자가 무료로 액세스할 수 있는 경우도 많습니다. 그 결과, WiFi 네트워크 연결이 안전하지 않을 수 있습니다. 하지만 특정 유형의 WiFi 연결은 더 안전합니다.

- 액세스가 제한된 개인 WiFi 네트워크에 연결  
회사나 대학에서 설정한 내부 네트워크와 같은 개인 네트워크는 올바른 보안 챌린지를 제공할 수 있는 사용자만 네트워크에 액세스할 수 있도록 제한합니다. 잠재적 사용자는 연결 시퀀스 도중 키를 제공하거나 보안 VPN을 통해 네트워크에 로그인해야 합니다.
- WiFi 네트워크에 대한 연결 암호화  
보안 키를 사용하여 시스템과 WiFi 네트워크 간의 통신을 암호화할 수 있습니다. WiFi 네트워크에 대한 액세스 포인트는 보안 키 생성 기능이 있는 홈 또는 사무실 라우터여야 합니다. 시스템과 라우터는 보안 연결을 만들기 전에 키를 설정하고 공유합니다.

dladm 명령은 액세스 포인트를 통한 연결의 암호화를 위해 WEP(Wired Equivalent Privacy) 키를 사용할 수 있습니다. WEP 프로토콜은 무선 연결에 대한 IEEE 802.11 사양에서 정의됩니다. dladm 명령의 WEP 관련 옵션에 대한 자세한 내용은 [dladm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

### ▼ 암호화된 WiFi 네트워크 연결을 설정하는 방법

다음 절차에서는 홈 시스템과 홈 라우터 간에 보안 통신을 설정하는 방법을 설명합니다. 많은 유무선 홈 라우터에는 보안 키를 생성할 수 있는 암호화 기능이 있습니다.

**시작하기 전에** 자신의 고유한 무선 네트워크에 연결하는 경우 라우터가 구성되었고 WEP 키가 생성되었는지 확인합니다. 키 구성을 생성 및 저장하는 방법은 라우터 제조업체의 설명서를 참조하십시오.

#### 1 관리자로 전환합니다.

자세한 내용은 [Oracle Solaris 11.1 관리: 보안 서비스의 “지정된 관리 권한을 사용하는 방법”](#)을 참조하십시오.

#### 2 WEP 키가 포함된 보안 객체를 만듭니다.

시스템에서 터미널 창을 열고 다음을 입력합니다.

```
# dladm create-secobj -c wep keyname
```

여기서 *keyname*은 키에 지정할 이름을 나타냅니다.

**3 보안 객체에 WEP 키의 값을 제공합니다.**

create-secobj 하위 명령이 키의 값을 요청하는 스크립트를 실행합니다.

```
provide value for keyname: 5-or-13-byte key
confirm value for keyname: Retype key
```

이 값은 라우터에서 생성된 키입니다. 스크립트는 ASCII나 16진수의 5바이트 또는 13바이트 문자열을 키 값으로 허용합니다.

**4 방금 만든 키의 콘텐츠를 확인합니다.**

```
# dladm show-secobj
OBJECT          CLASS
keyname         wep
```

여기서 *keyname*은 보안 객체의 이름입니다.

**5 WiFi 네트워크에 대해 암호화된 연결을 설정합니다.**

```
# dladm connect-wifi -e network -k keyname interface
```

**6 연결이 안전한지 확인합니다.**

```
# dladm show-wifi
LINK    STATUS    ESSID    SEC    STRENGTH  MODE  SPEED
net0    connected  wifi-1  wep    good      g     11Mb
```

SEC 제목 아래의 wep 값은 WEP 암호화가 연결에 적용되었음을 나타냅니다.

**예 5-3 암호화된 WiFi 통신 설정**

이 예에서는 이미 다음을 수행했다고 가정합니다.

- WEP 키를 만들 수 있는 홈 라우터에 시스템을 연결했습니다.
- 라우터 제조업체의 설명서에 따라 WEP 키를 만들었습니다.
- 시스템에 보안 객체를 만드는 데 사용할 수 있도록 키를 저장했습니다.

보안 객체를 만듭니다.

```
# dladm create-secobj -c wep mykey
provide value for mykey: *****
confirm value for mkey: *****
```

라우터에서 생성된 WEP 키를 제공할 때 입력한 값이 별표로 표시됩니다.

```
# dladm show-secobj
OBJECT          CLASS
mykey          wep
# dladm connect-wifi -e citinet -k mykey net0
```

앞의 명령은 보안 객체 *mykey*를 사용하여 WiFi 네트워크 *citinet*에 대한 암호화된 연결을 설정합니다.

```
# dladm show-wifi
LINK      STATUS      ESSID      SEC      STRENGTH  MODE      SPEED
net0      connected   citinet    wep      good      g         36Mb
```

이 출력 결과는 WEP 암호화를 통해 citinet에 연결되었음을 나타냅니다.



## 비교 맵: ifconfig 및 ipadm 명령

---

ipadm 명령은 네트워크 인터페이스 구성 목적으로 ifconfig 명령 대신 사용됩니다. ifconfig 명령은 Oracle Solaris 11에서도 계속 작동하지만 네트워크 구성을 위해 ipadm 명령을 사용하는 것이 더 좋습니다. 하지만 일부 ifconfig 옵션은 그에 상응하는 ipadm 하위 명령이 존재하지 않습니다. 다음 표에서는 ifconfig 명령에서 선택한 명령 옵션과 ipadm 명령의 동등한 옵션을 보여줍니다.

---

주 - ipadm 옵션의 전체 목록은 이 표에 제공되지 않습니다. 전체 목록은 [ipadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

---

표 A-1 ifconfig 및 ipadm 명령 간의 구문 매핑

ifconfig 명령	ipadm 명령
plumb/unplumb	ipadm create-ip ipadm create-vni ipadm create-ipmp ipadm enable-addr ipadm delete-ip ipadm delete-vni ipadm delete-ipmp ipadm disable-addr

표 A-1 ifconfig 및 ipadm 명령 간의 구문 매핑

(계속)

ifconfig 명령	ipadm 명령
[address[/prefix-length] [dest-address]] [addif address[ prefix-length]] [removeif address[ prefix-length]][netmask mask][destination dest-address]{auto-dhcp dhcp}[primary][wait seconds]extend   release   start	ipadm create-addr ipadm create-addr -T dhcp ipadm create-addr -T addrconf ipadm delete-addr ipadm refresh-addr
[deprecated   -deprecated] [preferred   -preferred] [private   -private] [zone zonetname   -zones   -all-zones][xmit   -xmit]	ipadm set-addprop ipadm reset-addprop ipadm show-addprop
up	ipadm up-addr
down	ipadm down-addr
[metric n] [mtu n] [nud   -nud] [arp   -arp] [usesrc [name   none] [router   -router]	ipadm set-ifprop ipadm show-ifprop ipadm reset-ifprop
[ipmp] [group [name   ""]] standby   -standby] [failover   -failover]	ipadm create-ipmp ipadm delete-ipmp ipadm add-ipmp ipadm remove-ipmp ipadm set-ifprop -p [standby] [group]
[interface] [-a]	ipadm ipadm show-if ipadm show-addr
[tdst tunnel-dest-addr] [tsrc tunnel-srcs-addr] [encaplimit n  -encaplimit] [thoplimit n]	dladm *-iptun 명령 세트. 자세한 내용은 dladm(1M) 매뉴얼 페이지와 <b>Oracle Solaris 11.1 네트워크 구성 및 관리</b> 의 “dladm 명령을 통한 터널 구성 및 관리”를 참조하십시오.
[auth_algs authentication algorithm] [encr_algs encryption algorithm] [encr_auth_algs encryption authentication algorithm]	ipseconf 자세한 내용은 ipseconf(1M) 매뉴얼 페이지 및 <b>Oracle Solaris 11.1의 네트워크 보안</b> 의 7장, “IPsec 구성(작업)”을 참조하십시오.

표 A-1 ifconfig 및 ipadm 명령 간의 구문 매핑 (계속)

ifconfig 명령	ipadm 명령
[auth_revarp] [ether <i>address</i> ] [index <i>if-index</i> ] [subnet <i>subnet-address</i> ] [broadcast <i>broadcast-address</i> ] [token <i>address /prefix-length</i> ]  DHCP 옵션 - inform, ping, release, status, drop	현재 사용할 수 있는 동등한 하위 명령은 없습니다.
[modlist] [modinsert <i>mod_name@ pos</i> ] [modremove <i>mod_name@pos</i> ]	현재 사용할 수 있는 동등한 하위 명령은 없습니다.



## 비교 맵: ndd 및 ipadm 명령

---

ipadm 명령은 네트워크 매개변수 또는 조정 가능 항목을 사용자 정의하기 위한 목적으로 ndd 명령 대신 사용됩니다. ndd 명령은 Oracle Solaris 11에서도 계속 작동하지만 네트워크 매개변수를 사용자 정의하기 위해서는 ipadm 명령을 사용하는 것이 더 좋습니다. 하지만 일부 ndd 옵션은 그에 상응하는 ipadm 하위 명령이 존재하지 않습니다. 다음 표에서는 ndd 명령에서 선택한 명령 옵션과 ipadm 명령의 동등한 옵션을 보여줍니다.

---

주 - ipadm 옵션의 전체 목록은 이 표에 제공되지 않습니다. 전체 목록은 [ipadm\(1M\)](#) 매뉴얼 페이지를 참조하십시오.

---

표 B-1 ndd 및 ipadm 명령 사이의 구문 매핑: 등록 정보 검색

ndd 명령	ipadm 명령
<pre>bash-3.2# ndd -get /dev/ip ? ip_def_ttl      (read and write) ip6_def_hops    (read and write) ip_forward_directed_broadcasts                 (read and write) ip_forwarding   (read and write) ... ...</pre>	<pre>bash-3.2# ipadm show-prop ip PROTO PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE ipv4 forwarding  rw    off      --         off         on,off ipv4 ttl         rw    255     --         255        1-255 ipv6 forwarding  rw    off      --         off         on,off ipv6 hoplimit    rw    255     --         255        1-255 ...</pre>
<pre>bash-3.2# ndd -get /dev/ip \ ip_def_ttl 100 bash-3.2# ndd -get /dev/ip \ ip6_def_hops 255</pre>	<pre>bash-3.2# ipadm show-prop -p ttl,hoplimit ip PROTO PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE ipv4 ttl         rw    255     --         255        1-255 ipv6 hoplimit    rw    255     --         255        1-255</pre>
<pre>bash-3.2# ndd -get /dev/tcp ? tcp_cwnd_max    (read and write) tcp_strong_iss  (read and write) tcp_time_wait_interval                 (read and write) tcp_tstamp_always (read and write) tcp_tstamp_if_wscale                 (read and write) ... ...</pre>	<pre>bash-3.2# ipadm show-prop tcp PROTO PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE tcp  ecn         rw    passive --         passive  never,passive, active tcp  extra_     rw    2049    2049,4045  2049,4045  1-65535 priv_ports tcp  largest_    rw    65535  --         65535     1024-65535 anon_port tcp  recv_       rw    128000 --         128000    2048-1073741824 maxbuf tcp  sack        rw    active --         active    never,passive, active tcp  send_       rw    49152  --         49152     4096-1073741824 maxbuf tcp  smallest_   rw    32768  --         32768     1024-65535 anon_port tcp  smallest_   rw    1024   --         1024      1024-32768 nonpriv_port ... ... ... bash-3.2# ipadm show-prop -p ecn,sack tcp PROTO PROPERTY  PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE tcp  ecn         rw    passive --         passive  never,passive,active tcp  sack        rw    active  --         active    never,passive,active</pre>

표 B-2 ndd 및 ipadm 명령 사이의 구문 매핑: 등록 정보 설정

ndd 명령	ipadm 명령
<pre>bash-3.2# ndd -set /dev/ip \ ip_def_ttl 64 bash-3.2# ndd -get /dev/ip \ ip_def_ttl 64</pre>	<pre>bash-3.2# ipadm set-prop -p ttl=64 ipv4 bash-3.2# ipadm show-prop -p ttl ip PROTO PROPERTY FAMILY PERM VALUE DEFAULT POSSIBLE ip   ttl      inet   rw   64   255   1-255 PROTO PROPERTY PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE ipv4  ttl      rw     64       64         255     1-255 bash-3.2# ipadm reset-prop -p ttl ip bash-3.2# ipadm show-prop -p ttl ip PROTO PROPERTY PERM  CURRENT  PERSISTENT  DEFAULT  POSSIBLE ipv4  ttl      rw     255     255         255     1-255</pre>



# 색인

---

## A

ARP(Address Resolution Protocol), 14  
autopush 등록 정보, 30

## B

BDP(대역폭 지연 곱), 48-49  
BSSID, 56

## C

cfgadm 명령, 36  
CIDR 표기법, 19

## D

DefaultFixed NCP, 11  
DHCP, 20  
dladm 명령, 11-14, 25-27  
  connect-wifi, 57  
  delete-phys, 27  
  help, 12-13  
  rename-link, 27  
  reset-linkprop, 28-33  
  scan-wifi, 56  
  set-linkprop, 28-33  
  show-ether, 32-33, 33  
  show-link, 26  
  show-linkprop, 32-33, 59  
  show-phys, 26-27

dladm 명령 (계속)

  show-wifi, 57  
DMA(직접 메모리 액세스), 31  
DR(동적 구성), NIC 교체, 35

## E

ECMP, 46-47  
ENM(외부 네트워크 수정자), 10  
ESSID, 56  
/etc/hosts 파일, 19

## G

GLD(일반 LAN 드라이버), 31-32  
GLDv3, 12-13

## I

ICMP, 14  
ifconfig 명령, 13  
  및 ipadm 명령, 65-67  
IP 인터페이스  
  IP 주소, 53, 54  
  IP 주소 변경, 40-41  
  IP 주소 제거, 40-41  
  IP 주소 지정, 19  
  IPMP 인터페이스, 18-22  
  MAC 주소 고유성 확인, 16-17  
  TCP/IP 프로토콜 등록 정보, 43-49

## IP 인터페이스 (계속)

- VNI 인터페이스, 18-22
- WiFi, 56
- 구성, 21
- 권한이 부여된 포트, 45
- 만들기 및 연결, 18-22
- 모니터링, 50-54
- 사용 안함 및 사용, 40
- 인터페이스 구성 삭제, 39-40
- 인터페이스 등록 정보, 52
- 인터페이스 등록 정보 표시, 41
- 주 인터페이스 변경, 39-40
- 주소 등록 정보, 42
- 패킷 전달 사용, 41-42, 44-45
- 표시
  - IP 주소, 53
  - 인터페이스, 51-52
  - 인터페이스 등록 정보, 52
  - 일반 정보, 20, 50-51
  - 주소 등록 정보, 54
  - 프로토콜 등록 정보, 44

## IP 주소

- DHCP, 20
- IPv4 및 IPv6, 19
- 등록 정보, 42, 54
- 로컬 및 원격, 19
- 모니터링, 50-54
- 인터페이스 등록 정보 설정, 41
- 정적, 19
- 제거, 40-41
- 패킷 전달, 41-42, 44-45

## IP 터널, 19

- 로컬 및 원격 주소, 19

## ipadm 명령, 11-14, 39-54

- create-addr, 19
- create-ip, 18-22
- delete-addr, 40-41
- delete-ip, 39-40
- disable-ip, 40
- help, 13, 14
- ifconfig 명령 비교, 65-67
- ndd 명령 비교, 69-71
- set-addrprop, 42
- set-ifprop, 41

## ipadm 명령 (계속)

- set-prop, 43-49
- show-addr, 53
- show-addrprop, 42, 54
- show-if, 51-52
- show-ifprop, 41, 52
- show-prop, 43-49
- IPMP(IP 다중 경로 지정 인터페이스), 18-22

## M

- MAC 주소, 고유성 확인, 16-17
- MTU, 28-29

## N

- name-service/switch 서비스, 19
- NCP, “네트워크 구성 프로파일”참조
- NCP(네트워크 구성 프로파일), 9-14
  - DefaultFixed, 10, 11
  - NCP 나열, 17-18
  - 고정된, 10-11
  - 반응적, 10-11
  - 활성 NCP, 9-14
  - 활성 NCP 전환, 11, 17-18

## nnd 명령, 14

- 및 ipadm 명령, 69-71

## netadm 명령, 11-14, 17-18

## netcfg 명령, 11-14

## netstat 명령, WiFi 링크를 통해 패킷 플로우 확인, 60

## NIC(네트워크 인터페이스 카드), 교체, DR 사용, 35

## NIC 드라이버, 28

## R

- route 명령, 21

## S

- SCTP, 14

SMF(서비스 관리 기능), 13  
 STREAMS 모듈, 및 데이터 링크, 30

## T

TCP 수신 버퍼 크기, 48-49

## U

UDP, 14  
 USENIX, 31-32

## V

VLAN(Virtual Local Area Network), 25-26  
 VNI(가상 네트워크 인터페이스), 18-22  
 VNIC(가상 네트워크 카드), 25-26  
 VPN(가상 사설망), 30

## W

### WiFi

BSSID(Basic Service Set ID), 56  
 ESSID(Extended Service Set ID), 56  
 IEEE 802.11 사양, 55  
 WiFi 구성 예제, 58  
 WiFi 네트워크에 연결, 56, 57, 58  
 링크 모니터링, 59  
 보안 WiFi 링크, 61  
 암호화된 통신 예, 62  
 연결 암호화, 61  
 예, 링크 속도 설정, 60  
 정의, 55  
 패킷 플로우 확인, 60

WLAN, 10

## 고

고정된 네트워크 구성, 9-14  
 정적 IP 주소, 10

## 권

권한이 부여된 포트, 45

## 네

네트워크 구성 도구, 11-14  
 dladm 명령, 12-13  
 ipadm 명령, 13  
 netadm 명령, 11  
 netcfg 명령, 11

## 대

대칭 경로 지정, 46-47

## 데

### 데이터 링크

autopush 등록 정보, 30  
 DMA 바인딩, 31  
 MTU 크기 변경, 28-29  
 STREAMS 모듈, 30  
 VLAN, 25-26  
 VNIC, 25-26  
 공용 및 개인 등록 정보, 28  
 링크 속도, 29-30  
 링크 집계, 25-26  
 링크를 통해 IP 인터페이스 구성, 18-22  
 물리적 링크, 26  
 속성 설정, 28-33  
 알려진 속도 및 사용으로 설정된 속도, 29-30  
 이더넷 매개변수 값, 32-33  
 이름 바꾸기, 27  
 인터럽트 속도, 31-32  
 일반 이름, 27  
 자동 협상, 28  
 제거, 27  
 표시  
 네트워크 드라이버 등록 정보, 32-33, 33  
 링크, 26  
 링크 등록 정보, 32-33  
 물리적 속성, 26-27

데이터 링크, 표시 (계속)  
시스템의 물리적 위치, 27  
일반 정보, 25-26

**로**  
로컬 주소, 19

**링**  
링크 속도, 29-30  
링크 이름, 27  
링크 집계, 25-26

**멀**  
멀티홈 호스트, 46-47

**무**  
무선 인터페이스, 55

**반**  
반응적 네트워크 구성, 9-14  
ENM, 10  
WLAN, 10  
위치 프로파일, 10  
반이중, 29

**버**  
버퍼, 31

**보**  
보안 고려 사항, WiFi, 61

**원**  
원격 주소, 19

**위**  
위치 프로파일, 10

**이**  
이더넷 매개변수, 32-33

**인**  
인터럽트 속도, 31-32

**자**  
자동 협상, 28

**전**  
전원 관리, 30  
전이중, 29

**점**  
점보 프레임, 지원 사용, 28-29

**주**  
주 인터페이스, 전환, 27, 33-37, 39-40  
주소 객체, 20

**패**  
패킷 전달  
인터페이스에, 41-42

패킷 전달 (계속)

프로토콜에, 44-45

**프**

프로토콜, 등록 정보, 43-49

프로파일 관리 네트워크 구성, 9-10

**혼**

혼잡 제어, 47-48

